Deteksi Lobster Menggunakan teknik StrongSORT pada YOLOv7

1st Ghanes Mahesa Aditya
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ganeshmahesa@student.telkomuniversity.ac.id

2nd Ledya Novamizanti
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ledyaldn@telkomuniversity.ac.id

3rd Fityanul Akhyar
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
fityanul@telkomuniversity.ac.id

***Abstrak* — Melakukan *object-tracking* dengan mempunyai akurasi dan performa yang tinggi merupakan hal penting dalam penerapan pemantauan pada *deep learning* yang diterapkan pada sebagai automatic driving, and intelligent monitoring salah satunya adalah *lobster monitoring*. Dalam mencapai hal tersebut diperlukan beberapa penelitian yang digabungkan menjadi satu dari mulai *object-detection*, dan *object-tracking*. Saat ini, dalam halnya *object-detection* ada beberapa algoritma yang sangat cukup popular salah satunya adalah yaitu YOLO dengan memiliki akurasi, dan kecepatan deteksi yang tinggi. Dengan berkembangnya zaman YOLO dilakukan peningkatan dengan menghasilkan YOLOv7 yang sangat canggih dari YOLO versi lainya, dengan memiliki akurasi tertinggi yaitu 56.8% dan 30 FPS. Maka dari itu YOLOv7 layak untuk diterapkan dalam *object-detection* yang akan digabungkan dengan *StrongSORT*. StrongSORT adalah *object-tracking* yang sangat kuat saat ini dengan meningkatkan beberapa sitem pada DeepSORT. Dengan menggabungkan dua sistem *deep learning* menjadi satu dan dilakukan pelatihan pada *dataset* lobster menghasilkan *frame per second* (FPS) diatas 4, dari penguji cobaan pada data percobaan didapatkan *precision* sebesar 0.90, *recall* mendapatkan nilai 0.81, mAP@0.5 yang mencapai 0.87 dan untuk mAP@0.5-0.95 tertinggi 0.44. Dari hasil yang didapatkan dapat disimpulkan bahwa sistem memenuhi syarat yang dapat dikatakan *real-time object detection*, dan *object tracking*.**

***Kata kunci*— object-detection, StrongSORT, YOLOv7, object-tracking**

I. PENDAHULUAN

Baru baru ini, *deep learning* salah metode dari *artificial intelligence* (AI) yang popular dalam pengolahan data dari mulai data structural hingga data unstructural, dengan melakukan pembuatan layer non liniear tersembunyi untuk ekstraksi fitur. Standar *neural network* (NN) terdiri dari banyak prosesor sederhana yang terhubung yang disebut neuron, masing-masing menghasilkan urutan aktivasi bernilai nyata. Neuron input diaktifkan melalui sensor yang memahami lingkungan, neuron lain diaktifkan melalui koneksi berbobot dari neuron yang sebelumnya aktif [1]. Salah satu, kegunaan yang sangat sering digunakan pada *deep learning* adalah pengolahan citra yaitu lebih sering dikenal sebagai computer vision dengan membantu banyak aspek-aspek seperti mengolah data lobster dengan melakukan metode multiple object tracking (MOT). Multiple object tracking (MOT) umumnya mengacu pada deteksi dan pelacakan ID beberapa target dalam video, seperti pejalan kaki, mobil, hewan, dll., tanpa mengetahui jumlah target terlebih dahulu [2].

YOLO [3] adalah salah satu algoritma yang dikembangkan pada area computer vision untuk melakukan deteksi objek, dan menghitung objek secara real-time yang dikenalkan pada tahun 2015. Menggunakan metode membingkai ulang deteksi objek sebagai masalah regresi tunggal, langsung dari piksel gambar ke koordinat kotak pembatas dan probabilitas kelas. Menggunakan sistem kami, *you only look once* (YOLO) pada gambar untuk memprediksi objek apa yang ada dan di mana mereka berada [3]. Perkembangan versi pada YOLO begitu berkembang pesat hingga saat ini dalam mendeteksi objek secara cepat, untuk saat ini versi model YOLO [3] yang performanya stabil adalah YOLOv7 [4] dalam kecepatan dan akurasi. Terdapat beberapa penelitian juga yang sangat membutuhkan *object detection* seperti klasifikasi ikan dalam air [5] dan sistem inspeksi cacat pada permukaan kayu [6], [7].

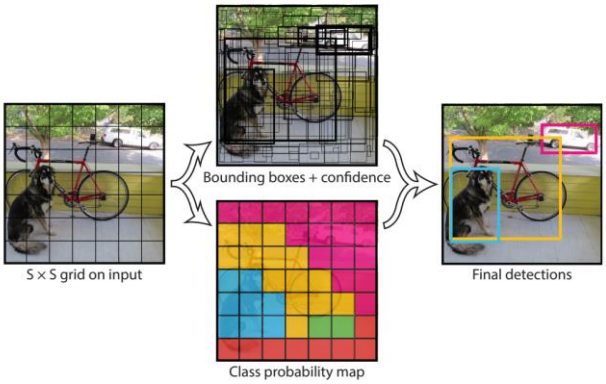
Dalam melakukan deteksi objek dan menghitung diperlukan teknik tambahan untuk dilakukan pelacakan objek, agar objek dapat terlacak salah satu metode yang saat ini terkenal adal SORT yang dikembangkan menjadi StrongSORT. Simple Online and Realtime Tracking (SORT) adalah pendekatan pragmatis untuk pelacakan beberapa objek dengan fokus pada algoritme yang sederhana dan efektif. Selanjutnya, metode SORT terdiri dari tiga langkah yaitu menggunakan deteksi Faster Region CNN *detection* framework [8], estimasi model menggunakan Kalman Filter [9] , dan kelas yang menentukan menggunakan Hungarian Algorithm [10] [11]. Pengembangan dilakukan pada SORT dengan menghasilkan DeepSORT. DeepSort is an object tracking method that combines object detection with probabilistic data association (PDA) based tracking techniques[12]. DeepSORT menggunakan informasi deteksi objek untuk memperkirakan lokasi objek dalam bingkai berikutnya dan pembaruan yang memperkirakan setiap kali objek terdeteksi. Selanjutnya DeepSORT dikembangkan kembali menjadi StrongSORT. StrongSORT adalah algoritma untuk pelacakan objek juga, dan peningkatan dari DeepSORT atau bisa dibilang DeepSORT yang Lebih Kuat. Keunggulan StrongSORT dibandingkan DeepSORT adalah IDFI, HOTA, dan MOTA. DeepSORT sendiri memiliki IDFI 77.3, *multiple object tracking accuracy* (MOTA) 76.7, dan HOTA 69.6, sedangkan StrongSORT memiliki IDFI 82.3,

MOTA 77.1, HOTA 69 [13]. Dari segi performa, StrongSORT lebih baik dari DeepSORT. Dengan menggambung deteksi objek, perhitungan deteksi, dan pelacakan beberapa objek menggunakan StrongSORT dapat menghasilkan perhitungan yang akurat pada dataset lobster.

II. KAJIAN TEORI

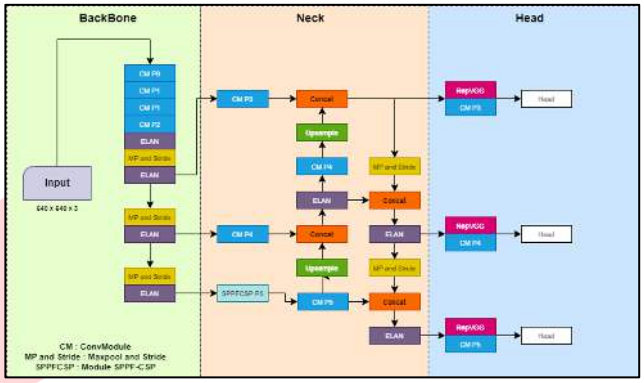
A. YOLOv7

Sistem YOLOv7 membagi gambar input menjadi kisi-kisi S × S. Jika pusat objek jatuh ke dalam sel kisi, sel kisi tersebut bertanggung jawab untuk mendeteksi objek tersebut. Setiap sel kisi memprediksi kotak pembatas B dan skor kepercayaan untuk kotak tersebut. Skor kepercayaan ini mencerminkan seberapa yakin model bahwa kotak itu berisi objek dan juga seberapa akurat menurut prediksi kotak itu. Setiap kotak pembatas terdiri dari 5 prediksi yaitu x, y, w, h, dan konfidensi. Koordinat (x, y) mewakili pusat kotak relatif terhadap batas-batas sel grid [3]. Ilustrasi cara kerja YOLO[3] dapat dilihat pada Gambar 1. Saat ini, YOLO sudah dikembangkan menjadi model yang lebih cepat dan keakuratanya lebih tinggi dari YOLO sebelumnya, YOLO yang sangat popular dan stabil saat ini adalah YOLOv7. YOLOv7 mengusulkan arsitektur baru dari detektor objek waktu nyata dan metode penskalaan model yang sesuai. Selain itu, kami menemukan bahwa proses metode deteksi objek yang berkembang menghasilkan topik penelitian baru. Selama proses penelitian, kami menemukan masalah penggantian modul yang diparameterisasi ulang dan masalah alokasi penetapan label dinamis. Untuk mengatasi masalah tersebut, kami mengusulkan metode bag-of-freebies yang dapat dilatih untuk meningkatkan akurasi deteksi objek. Berdasarkan hal di atas, kami telah mengembangkan YOLOv7 serangkaian sistem deteksi objek, yang menerima hasil mutakhir [4].



Gambar 1. Deteksi model sistem kami sebagai masalah regresi. Ini membagi gambar menjadi kisi S × S dan untuk setiap sel kisi memprediksi kotak pembatas B, kepercayaan untuk kotak tersebut, dan probabilitas kelas C. Prediksi ini dikodekan sebagai S × S × (B ∗ 5 + C) tensor [3].

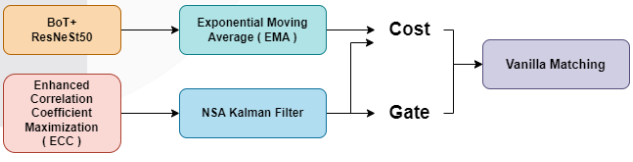
Perkembangan terkini dalam arsitektur deteksi objek YOLOv7[4], telah melibatkan transformasi berupa pengenalan arsitektur E-ELAN dan pendekatan penskalaan untuk model yang berfokus pada penggabungan. Arsitektur model ini terbagi menjadi tiga komponen utama: *backbone, neck*, dan *head*, masing-masing dengan peran dan fungsi yang berbeda-beda. Komponen *backbone* bertugas untuk mengekstraksi fitur-fitur signifikan dari gambar dan meneruskannya ke komponen *head* melalui *neck*. Sementara itu, *neck* bertugas mengumpulkan peta fitur yang telah diekstraksi oleh komponen *backbone* dan menghasilkan piramida fitur yang memungkinkan analisis lebih mendalam [3]. Ilustrasi visual dari komponen-komponen ini dapat diamati dalam Gambar 2, memberikan pemahaman yang lebih jelas mengenai susunan arsitektur yang digunakan.



GAMBAR 2.
Arsitektur YOLOv7

B. StrongSORT

StrongSORT mengunjungi kembali pelacak klasik DeepSORT dan memutakhirkannya dengan modul baru dan beberapa trik inferensi. Pelacak baru yang dihasilkan, StrongSORT, dapat berfungsi sebagai dasar kuat baru untuk tugas multiple object tracking (MOT) . Dan juga, usulkan dua algoritme yang ringan dan bebas tampilan, AFLink dan GSI, untuk memecahkan masalah asosiasi yang hilang dan deteksi yang hilang. Eksperimen menunjukkan bahwa mereka dapat diterapkan dan bermanfaat bagi berbagai pelacak canggih dengan biaya komputasi tambahan yang dapat diabaikan [13]. Peningkatan kami atas DeepSORT mencakup modul lanjutan yang menggantikan Faster R-CNN [8] dengan YOLO-X [14], *exponential moving averages* (EMA) mengurangi sensitivitas dari kebisingan deteksi, *enhanced correlation coefficient maximization* (ECC) [15] model untuk *camera motion compensation*. Arsitektur pada StrongSORT dapat dilihat pada Gambar 3.

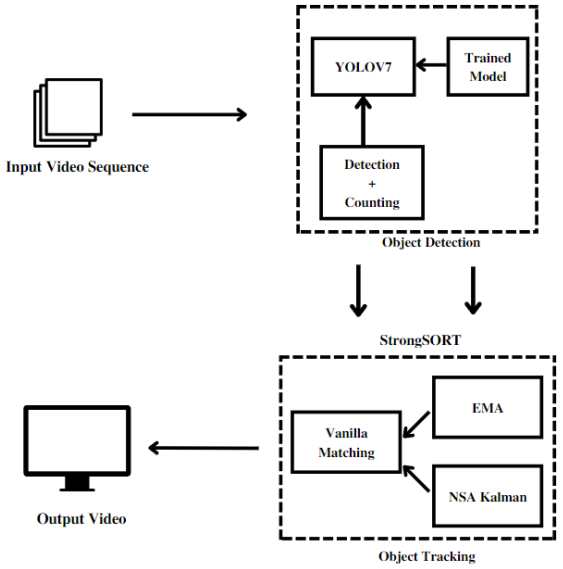


GAMBAR 3.
Arsitektur StrongSORT

C. YOLOv7-StrongSORT

Dapat diperhatikan bahwa hasil dari proses *object-detection* menggunakan YOLOv7[4] menunjukkan tingkat performa yang sangat baik, dengan kemampuan *object-tracking* yang secara konsisten memiliki nilai yang tinggi yaitu StrongSORT[13]. Menggabungkan dua sistem tersebut menjadi YOLOv7-StrongSORT. Proses operasional yang dijalankan dapat ditemukan dalam rincian yang disajikan pada Gambar 3. Proses pelatihan untuk kedua model

dilakukan secara terpisah. YOLOv7 menjalankan tahap pelatihan menggunakan *dataset* yang telah tersedia, dengan proses yang dilakukan melalui *platform* Google Collab. Sementara itu, untuk StrongSORT mengambil manfaat dari pendekatan berbasis model terbaik untuk melalui proses pelatihan.



GAMBAR 3.
Proses dari YOLOv7-StrongSORT. Input didapatkan dari pemecahan suatu video menjadi beberapa bagian *frame*

III. DATASET DAN SPESIFIKASI

A. Dataset

Data yang merupakan bahan dasar dalam proses pelatihan diperoleh dari berbagai video dan foto yang dikumpulkan dari berbagai sumber di internet. Selanjutnya, data ini telah diolah hingga menghasilkan kumpulan sebanyak 176 foto. Melalui penerapan augmentasi berupa metode *flip*, jumlah foto berhasil diperluas menjadi 367 foto yang kemudian secara proporsional terbagi menjadi tiga kelompok, masing-masing dengan perbandingan 93% : 0.05% : 0.03% untuk tahap pelatihan, validasi, dan pengujian. Dengan rincian spesifik, terdapat 340 foto dalam kelompok pelatihan, 17 foto dalam kelompok validasi, serta 10 foto dalam kelompok pengujian. Selanjutnya, untuk memastikan konsistensi dan standar kualitas data, dilakukan pengubahan resolusi gambar menjadi 640×640 piksel sebelum tahap anotasi dilaksanakan, dimana langkah ini dapat diikuti dengan referensi lebih jauh pada Gambar 3 (a-b).

(a)

(b)

GAMBAR 4.
Dataset (a) Gambar Hasil Annotasi; (b) Gambar Orisinil

Proses anotasi dalam hal ini dilakukan dengan menerapkan pendekatan *bounding box* yang telah diadaptasi sesuai dengan kebutuhan model, khususnya model YOLOv7. Hasil dari anotasi ini dengan jelas dapat ditemukan dalam Gambar 4(a), yang memberikan representasi visual tentang penerapan *bounding box* pada objek-objek yang teridentifikasi dalam dataset. Pendekatan anotasi ini memungkinkan untuk mengakomodasi kekhasan dan kebutuhan model YOLOv7 secara optimal. Melalui Langkah ini memastikan bahwa anotasi yang dihasilkan sesuai dengan kerangka kerja deteksi objek yang diadopsi dalam model tersebut. Dengan demikian, hasil anotasi ini menjadi langkah kunci dalam mempersiapkan data yang sesuai dengan karakteristik deteksi objek yang diinginkan.

B. Spesifikasi

Dalam proses melakukan pengujian dilakukan dengan menggunakan laptop pribadi yaitu laptop Lenovo Legion 5. Laptop ini ditenagai processor AMD Ryzen 7 4800H dengan grafis GPU NVIDIA GeForce GTX 1650Ti 4GB yang sudah terpasang CUDA dan cuDNN. CUDA membantu pengembang untuk mengakses daya komputasi mentah GPU CUDA untuk memproses data lebih cepat dibandingkan dengan CPU tradisional[16]. Untuk proses pelatihan dibantu menggunakan The Google Colab layanan menyediakan 12,72 GB RAM dan 358,27 GB ruang hard disk dalam satu runtime. Setiap runtime berlangsung selama ± 6 jam setelah runtime diatur ulang dan pengguna harus membuat koneksi lagi, dan menggunakan mesin GPU Tesla T4.

Evaluasi terhadap hasil optimal sangat difokuskan pada dua aspek utama, yaitu *mean average precision* (mAP) dan

frame per seconds (FPS). mAP mengacu pada rata-rata *average precision,* di mana dalam beberapa konteks, setiap kelas memiliki nilai *average precision* yang dihitung dan kemudian diambil rata-ratanya [14]. Sementara itu, FPS adalah satuan yang mengukur performa perangkat tampilan dalam merekam video. Dengan demikian, pengujian model dapat dievaluasi dari dua perspektif, yaitu sejauh mana kualitas mAP yang tercermin dalam persamaan (1) dan sejauh mana FPS yang menunjukkan keseimbangan antara akurasi, kecepatan, dan presisi dalam penerapan model pada proses perekaman video, sebagaimana tercermin dalam persamaan (2). Kombinasi penilaian ini memberikan gambaran komprehensif tentang performa dan responsivitas model dalam konteks deteksi objek *real-time*.

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \quad (1)$$

$$FPS = \frac{Frames}{(\,current\ time - start\ time)} \quad (2)$$

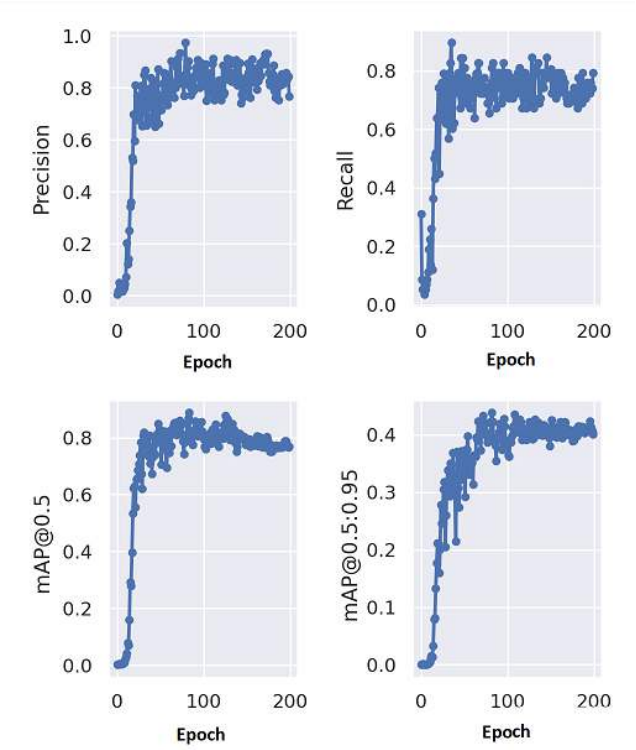
Evaluasi selanjutnya difokuskan pada pengukuran tingkat presisi dan *recall*, yang bertujuan untuk menilai sejauh mana prediksi positif yang dihasilkan oleh model secara konsisten sesuai dengan data aktual. *Recall* mencerminkan proporsi kasus prediksi positif yang berhasil diidentifikasi dengan benar sebagai positif. Pengukuran ini dilakukan dengan mempertimbangkan sejauh mana aturan TP (*true positive*) mencakup kasus prediksi positif. Dalam mengukur cakupan dari kasus prediksi positif melalui aturan TP, upaya dilakukan untuk memahami seberapa banyak kasus yang relevan yang telah tertangkap oleh aturan TP [15]. Untuk menggambarkan konsep ini lebih konkret, persamaan yang mendasari presisi dan *recall* dapat ditemukan dalam persamaan 1 dan persamaan 2. Evaluasi ini memberikan wawasan lebih dalam tentang kemampuan model dalam menghasilkan prediksi yang akurat dan konsisten sesuai dengan data aktual.

$$Precision = \frac{True\ Positive}{(\,False\ Positive - True\ Positive)} \quad (3)$$

$$Recall = \frac{True\ Positive}{(\,False\ Negative - True\ Positive)} \quad (4)$$

III. HASIL DAN PEMBAHASAN

Pada segmen ini, kami akan menjelaskan hasil yang diperoleh dari tahap pelatihan yang dilaksanakan pada model YOLOv7[2] menggunakan platform Google Collab. Proses pelatihan ini melibatkan pelaksanaan 200 epochs untuk mencapai hasil yang optimal. Detail dari langkah-langkah yang terlibat dalam proses pelatihan ini dapat ditemukan dalam Gambar 4, yang menggambarkan visualisasi berbagai metrik evaluasi seperti hasil mAP (Mean Average Precision), Precision, dan Recall yang terkait dengan proses pelatihan tersebut. Sumber data yang diolah dalam tahap ini diperoleh dari berbagai sumber internet, dengan total 176 foto untuk kemudian dilakukan proses augmentasi. Penerapan augmentasi berupa metode flip, jumlah foto berhasil diperluas menjadi 367 foto. Tahap ini menjadi sangat penting karena mempersiapkan dataset pelatihan yang beragam dan mewakili berbagai situasi dan kondisi. Tahap pelatihan ini memiliki peran sentral dalam membentuk model yang memiliki kemampuan untuk mendeteksi objek secara andal dan responsif, tak terkecuali dalam menghadapi variasi skenario yang mungkin terjadi.



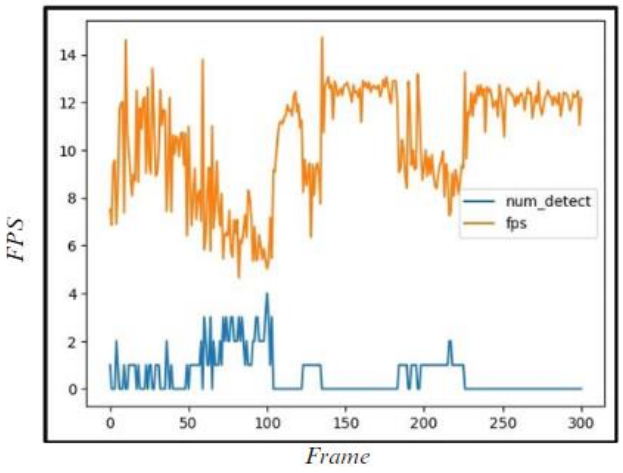
GAMBAR 4.
Hasil mAP, *Precision,* dan *Recall* pada Pelatihan di Google Collab

Dari hasil evaluasi yang telah dijalankan, diperoleh nilai precision sebesar 0.98, recall mendapatkan nilai 0.87, mAP@0.5 yang mencapai 0.89 dan untuk mAP@0.5-0.95 tertinggi 0.49 pada proses pelatihan model terbaik diambil berdasarkan mAP@0.5-0.95 dikarenakan lebih mengfokuskan akurasi pada *intersection over union* (IoU) dengan akurasi yang lebih tinggi. Model yang didapatkan akan dilakukan proses penguji cobaan pada data percobaan sebanyak 17 label, dengan hasil yang dapat terlihat pada Tabel 1. Dengan pencapaian hasil yang memuaskan, dapat disimpulkan bahwa proses pelatihan ini berjalan dengan sukses dan berhasil mencapai target yang telah ditetapkan. Evaluasi ini menegaskan bahwa model telah mencapai standar yang diharapkan.

TABEL 1.
Hasil mAP, *Precision,* dan *Recall* pada Data Percobaan

Parameter	Hasil
Precision	0.90
Recall	0.81
mAP@0.5	0.87
mAP@0.5-0.95	0.44

Langkah selanjutnya adalah mengintegrasikan dan menguji model yang telah dihasilkan pada video yang diambil dari sumber internet. Selama proses pengujian ini, perhatian khusus akan difokuskan pada pemantauan kecepatan pemrosesan *frame per seconds* (FPS), yang grafiknya dapat diidentifikasi dalam Gambar 5. Grafik tersebut merupakan hasil dari sistem yang dapat diartikan sebagai deteksi objek secara *real-time*. Dalam konteks deteksi objek secara *real-time*, nilai FPS yang mampu dihasilkan oleh model dapat mencapai rentang sekitar 3–5 FPS, dan dalam beberapa situasi bahkan mencapai 10 FPS. *Real-time object detection* dengan kecepatan tersebut sudah cukup tergantung pada karakteristik aplikasi [17].



GAMBAR 5.
Hasil FPS pada 300 *frames*

IV. KESIMPULAN

Tugas akhir ini mengusulkan untuk menggabungan dua sistem, yaitu YOLOv7 sebagai sistem deteksi objek, dan StrongSORT sebagai sistem pelacakan objek, dengan mendapatkan hasil yang menunjukkan bahwa kedua sistem ini berjalan dengan baik, dengan mampu menghasilkan FPS (*frames per second*) yang melebihi 4, serta untuk pengujian menggunakan data percobaan yang sudah ditentukan mendapatkan nilai *precision* sebesar 0.90 , *recall* sebesar 0.81, mAP@0.5 sebesar 0.87, mAP@0.5-0.95 sebesar 0.44, dengan itu hasil penelitian ini menunjukkan potensi besar dari sistem gabungan YOLOv7-StrongSORT dalam memberikan solusi efektif, dan efisien dalam deteksi serta pelacakan objek dalam aplikasi.

REFERENSI

[1] J. Schmidhuber, “Deep Learning in Neural Networks: An Overview,” Apr. 2014, doi: 10.1016/j.neunet.2014.09.003.

[2] F. Yang, X. Zhang, and B. Liu, “Video object tracking based on YOLOv7 and DeepSORT,” Jul. 2022.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” Jun. 2015.

[4] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” Jul. 2022.

[5] H. M. Lathifah, L. Novamizanti, and S. Rizal, “Fast and Accurate Fish Classification from Underwater Video using You Only Look Once,” *IOP Conf Ser Mater Sci Eng*, vol. 982, no. 1, p. 012003, Dec. 2020, doi: 10.1088/1757-899X/982/1/012003.

[6] F. Akhyar, L. Novamizanti, and T. Riantiarni,, “Sistem Inspeksi Cacat pada Permukaan Kayu menggunakan Model Deteksi Obyek YOLOv5,” *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 10, no. 4, p. 990, Oct. 2022, doi: 10.26760/elkomika.v10i4.990.

[7] F. Akhyar, L. Novamizanti, T. Putra, E. N. Furqon, M.-C. Chang, and C.-Y. Lin, “Lightning YOLOv4 for a Surface Defect Detection System for Sawn Lumber”, Accessed: Aug. 22, 2023. [Online]. Available: https://github.com/AlexeyAB/darknet

[8] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” Jun. 2015.

[9] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960, doi: 10.1115/1.3662552.

[10] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, no. 1–2, pp. 83–97, Mar. 1955, doi: 10.1002/NAV.3800020109.

[11] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple Online and Realtime Tracking,” Feb. 2016, doi: 10.1109/ICIP.2016.7533003.

[12] “Understanding Multiple Object Tracking using DeepSORT.” https://learnopencv.com/understanding-multiple-object-tracking-using-deepsort/ (accessed Jun. 24, 2023).

[13] Y. Du *et al.*, “StrongSORT: Make DeepSORT Great Again,” Feb. 2022.

[14] Y. Zhang *et al.*, “ByteTrack: Multi-Object Tracking by Associating Every Detection Box,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13682 LNCS, pp. 1–21, Oct. 2021, doi: 10.1007/978-3-031-20047-2_1.

[15] G. D. Evangelidis and E. Z. Psarakis, “Parametric image alignment using enhanced correlation coefficient maximization,” *IEEE Trans Pattern Anal Mach Intell*, vol. 30, no. 10, pp. 1858–1865, 2008, doi: 10.1109/TPAMI.2008.113.

[16] “Understanding NVIDIA CUDA: The Basics of GPU Parallel Computing.” https://www.turing.com/kb/understanding-nvidia-cuda (accessed Jun. 26, 2023).

[17] J. Lee and K. il Hwang, “YOLO with adaptive frame control for real-time object detection applications,” *Multimed Tools Appl*, vol. 81, no. 25, pp. 36375–36396, Oct. 2022, doi: 10.1007/S11042-021-11480-0/FIGURES/12.