Department of Digital Business
Journal of Artificial Intelligence and Digital Business (RIGGS)
Homepage: https://journal.ilmudata.co.id/index.php/RIGGS
Vol. 4 No. 4 (2025) pp: 12-19
P-ISSN: 2963-9298, e-ISSN: 2963-914X

Analisis Perbandingan Arsitektur dan Optimizer YOLOv11 untuk Estimasi Buah Kelapa

I Dewa Gede Agung Wibhisana Natih[1], Made Windu Antara Kesiman[2], I Made Gede Sunarya[3]
[1,2,3]Program Studi Pendidikan Teknik Informatika, Fakultas Teknik dan Kejuruan, Universitas Pendidikan Ganesha
[1]agung.wibhisana@undiksha.ac.id, [2]antara.kesiman@undiksha.ac.id, [3]sunarya@undiksha.ac.id

Abstrak

Estimasi jumlah buah kelapa dalam praktik Majeg di Bali secara manual dinilai tidak efisien dan sangat bergantung pada subjektivitas pengamat. Perkembangan teknologi computer vision, khususnya algoritma You Only Look Once (YOLO), menawarkan solusi deteksi objek yang cepat dan akurat untuk mengatasi keterbatasan tersebut. Namun, pemilihan arsitektur dan optimizer yang sesuai menjadi faktor penting dalam menentukan kinerja model secara keseluruhan. Penelitian ini bertujuan untuk menganalisis dan membandingkan sembilan kombinasi model yang terdiri atas tiga varian arsitektur YOLOv11 (n, s, m) dengan tiga optimizer berbeda (SGD, Adam, dan AdamW) guna menemukan konfigurasi paling optimal untuk tugas estimasi buah kelapa. Dataset penelitian mencakup 254 citra buah kelapa yang diambil di Desa Tusan, Klungkung, Bali, menggunakan kamera smartphone dengan dua kelas objek, yaitu kelapa muda dan kelapa tua. Proses pelatihan dilakukan menggunakan platform Google Colaboratory dengan akselerator GPU Tesla T4, sementara evaluasi kinerja dilakukan berdasarkan tiga metrik utama: akurasi (mAP50), kecepatan inferensi (FPS), dan ukuran model (Size). Analisis cost–benefit dengan bobot 50%–30%–20% menunjukkan bahwa YOLOv11n–SGD menjadi konfigurasi paling optimal dengan skor akhir 0.839, menawarkan keseimbangan terbaik antara akurasi tinggi (mAP50 0.926) dan efisiensi komputasi (5.5 MB). Hasil ini menunjukkan potensi penerapan model untuk estimasi buah kelapa secara real-time di lapangan menggunakan perangkat dengan sumber daya terbatas.).

Kata kunci : YOLO, Estimasi Buah Kelapa, Analisis Cost–Benefit, Optimizer, Computer Vision.

1. Latar Belakang

Kelapa (Cocos nucifera) memegang peranan vital dalam lanskap pertanian dan ekonomi Indonesia, negara yang diakui sebagai produsen kelapa terbesar di dunia. Lebih dari sekadar komoditas ekonomi, di Bali, kelapa terjalin erat dengan tatanan budaya dan spiritual masyarakat, menjadi elemen penting dalam berbagai upacara yadnya [1]. Keunikan interaksi sosial-ekonomi ini melahirkan sistem perdagangan tradisional yang dikenal sebagai majeg [2], sebuah praktik jual beli hak panen kelapa. Praktik serupa, yang dikenal sebagai sistem tebasan, borongan, atau ijon [2, 3], juga ditemukan dalam jual beli hasil pertanian lain seperti padi [2] dan petai [3], di mana hasil panen dibeli saat masih di pohon atau bahkan sebelum matang.

Meskipun memiliki akar sejarah yang panjang, praktik majeg [2] dihadapkan pada tantangan signifikan dalam era modern, terutama terkait proses estimasi jumlah buah kelapa di pohon. Metode tradisional yang mengandalkan pengamatan visual secara manual terbukti memiliki keterbatasan inheren. Subjektivitas pengamat, kesulitan visual akibat ketinggian pohon dan halangan daun, serta inefisiensi waktu menjadi kendala utama. Ketidakakuratan dalam estimasi ini—sebuah bentuk gharar atau ketidakpastian yang juga dikaji dalam konteks jual beli tebasan lainnya [2, 3]—tidak hanya berpotensi menimbulkan kerugian finansial, tetapi juga dapat mengancam keberlanjutan tradisi majeg itu sendiri.

Dalam upaya mengatasi keterbatasan metode manual, kemajuan teknologi computer vision [4] menawarkan solusi yang menjanjikan. Algoritma deep learning [5] seperti You Only Look Once (YOLO) [6] menjadi solusi yang menawarkan keseimbangan antara kecepatan pemrosesan real-time dan tingkat akurasi deteksi yang tinggi [7]. Prinsip dasar YOLO adalah membingkai deteksi objek sebagai masalah regresi, memprediksi kotak pembatas (bounding box) dan probabilitas kelas secara langsung dari citra penuh dalam satu evaluasi tunggal [6]. Efektivitas

YOLO dalam berbagai tugas deteksi objek telah dibuktikan dalam banyak penelitian, termasuk dalam konteks budaya seperti deteksi karakter wayang kulit Bali [7].

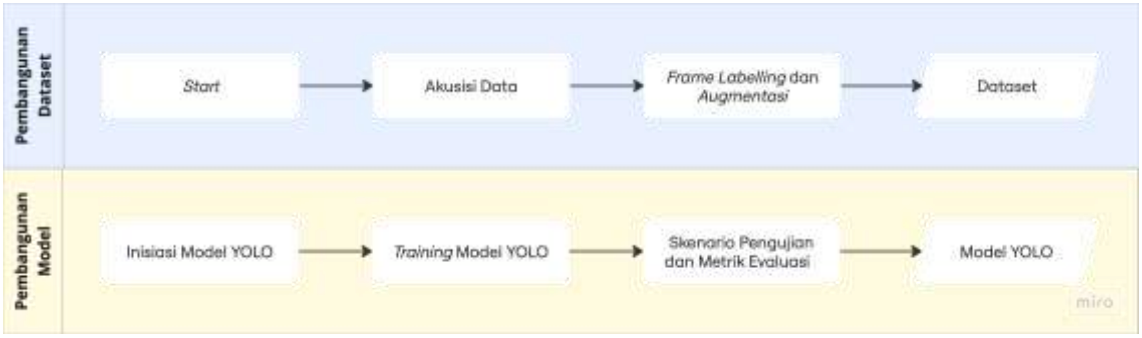
Namun, popularitas YOLO juga menghadirkan kompleksitas dalam implementasi praktis. Keluarga algoritma YOLO terus berkembang, menghasilkan berbagai varian arsitektur (misal: nano, small, medium, large seperti pada YOLOv5 [7]) dan berbagai algoritma optimisasi (optimizer) yang digunakan selama proses pelatihan, seperti Stochastic Gradient Descent (SGD), Adam, dan AdamW [4, 5]. Memilih kombinasi arsitektur dan optimizer yang tepat menjadi krusial, karena model dengan akurasi tertinggi (seringkali model yang lebih besar) mungkin tidak praktis untuk penerapan di lapangan karena tuntutan sumber daya komputasi yang tinggi dan kecepatan inferensi yang lebih lambat [7]. Analisis hyperparameter seperti ukuran citra, jumlah epoch pelatihan, dan ukuran batch juga terbukti signifikan dalam mempengaruhi hasil akhir model CNN [4].

Menyadari tantangan ini, penelitian ini secara spesifik berfokus pada analisis trade-off antara biaya (ukuran model, kecepatan komputasi) dan manfaat (akurasi deteksi) untuk menemukan konfigurasi model YOLOv11 yang paling optimal dalam konteks estimasi buah kelapa. Secara rinci, tujuan penelitian ini adalah:

1. Menguji dan membandingkan secara kuantitatif performa tiga varian arsitektur YOLOv11 (n, s, m) ketika dilatih menggunakan tiga optimizer yang umum digunakan (SGD, Adam, AdamW).
2. Menganalisis hasil perbandingan tersebut untuk mengidentifikasi konfigurasi arsitektur-optimizer mana yang menawarkan keseimbangan terbaik antara akurasi deteksi buah kelapa dan efisiensi komputasi (kecepatan inferensi dan ukuran model).

2. Metode Penelitian

Penelitian ini dilakukan melalui dua tahapan inti: (1) Pembangunan Dataset, dan (2) Pembangunan dan Evaluasi Model



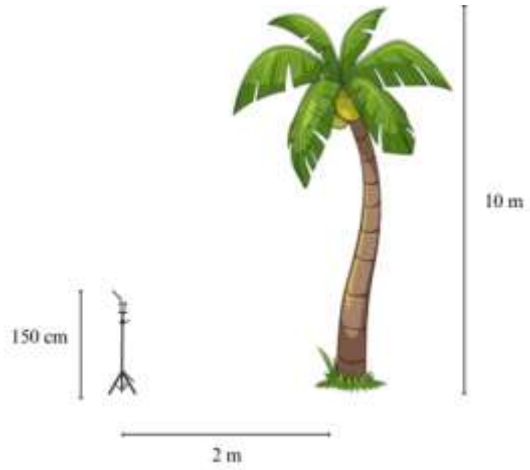
Gambar 1 Flowchart Tahapan Penelitian

2.1. Pembangunan Dataset

Kualitas dan representasi dataset merupakan fondasi fundamental bagi keberhasilan model machine learning, terutama dalam tugas computer vision [5, 8]. Oleh karena itu, tahap ini dilakukan dengan cermat melalui langkah-langkah berikut:

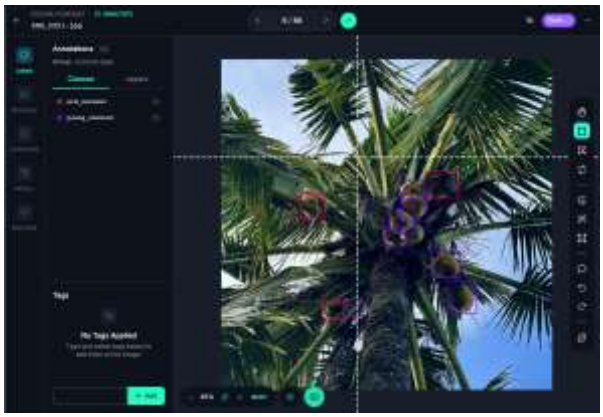
- Akuisisi Data: Pengumpulan data citra primer dilakukan menggunakan kamera smartphone (iPhone 14) untuk mereplikasi skenario penggunaan perangkat yang umum tersedia. Lokasi akuisisi difokuskan pada area perkebunan kelapa di Desa Tusan, Klungkung, Bali. Pengambilan gambar dilakukan pada sore hari (rentang waktu 15.00-18.00 WITA) untuk mendapatkan kondisi pencahayaan yang relatif konsisten. Parameter pengambilan gambar distandarisasi dengan menjaga jarak sekitar 2 meter dari batang pohon dan sudut elevasi kamera sekitar 77 derajat, bertujuan untuk menangkap area mahkota pohon tempat buah kelapa terkonsentrasi yang diilustrasikan pada Gambar 2.

DOI: https://doi.org/10.31004/riggs.v4i4.3329



Gambar 2 Ilustrasi Akusisi Data dengan Ponsel

- Pelabelan (Ground Truth Labelling): Setiap citra yang diperoleh kemudian diproses untuk anotasi objek. Platform Roboflow dipilih sebagai alat bantu untuk menggambar kotak pembatas (bounding box) secara manual di sekeliling setiap buah kelapa yang terlihat dalam citra. Proses ini serupa dengan yang dilakukan dalam pembuatan dataset untuk deteksi wayang [7] atau klasifikasi daun herbal . Dua kelas objek didefinisikan: kelapa_muda (diberi label 0) dan kelapa_tua (diberi label 1) sepeerti pada Gambar 3. Hasil anotasi ini, yang merepresentasikan data ground truth (kebenaran dasar), disimpan dalam format teks (.txt) yang sesuai dengan standar input YOLO.



Gambar 3 Pelabelan Menggunakan Roboflow

- Preprocessing dan Finalisasi Dataset: Citra yang telah dianotasi selanjutnya di-preprocess. Langkah pertama adalah mengubah ukuran (resize) semua citra menjadi resolusi 640x640 piksel. Ukuran ini dipilih karena umum digunakan sebagai input standar untuk banyak arsitektur YOLO, termasuk varian YOLOv5 [9], dan membantu dalam standardisasi input model. Selanjutnya, dataset dibagi menjadi tiga subset: data pelatihan (training), data validasi (validation), dan data pengujian (testing). Pembagian ini menggunakan metode Multilabel Stratified Shuffle Split untuk memastikan bahwa proporsi kelas (kelapa_muda dan kelapa_tua) terjaga secara seimbang di ketiga subset, mencegah bias selama pelatihan dan evaluasi. Rasio pembagian yang diadopsi adalah 80% untuk data pelatihan (204 gambar, 2164 anotasi), 10% untuk data validasi (26 gambar, 295 anotasi), dan 10% untuk data pengujian (24 gambar, 239 anotasi). Rasio ini merupakan praktik umum dalam pengembangan model machine learning [5, 8, 10]. Dataset final terdiri dari total 254 gambar unik dengan 2.698 anotasi buah kelapa. Perlu dicatat bahwa teknik augmentasi data, seperti flipping (membalik citra), rotasi acak, penyesuaian brightness dan kontras, serta penambahan noise (seperti Gaussian noise), merupakan praktik umum untuk meningkatkan jumlah dan variasi data pelatihan secara artifisial, yang seringkali dapat meningkatkan kemampuan generalisasi model [5, 8, 11]. Namun, dalam rangkaian eksperimen utama penelitian ini, augmentasi data tidak diterapkan secara eksplisit untuk menjaga fokus pada perbandingan arsitektur dan optimizer.

DOI: https://doi.org/10.31004/riggs.v4i4.3329

2.2. Pembangunan dan Evaluasi Model

Tahap ini dirancang untuk mengevaluasi secara sistematis dampak pilihan arsitektur dan optimizer terhadap kinerja model YOLOv11 pada tugas deteksi buah kelapa.

- Konfigurasi Eksperimen: Sembilan skenario eksperimen dirancang dengan mengkombinasikan tiga varian arsitektur YOLOv11 (yolo11n - nano, yolo11s - small, yolo11m - medium) dengan tiga algoritma optimizer (SGD, Adam, AdamW). Setiap kombinasi (misalnya, yolo11n-SGD, yolo11n-Adam, dst.) dilatih secara terpisah.

- Lingkungan Pelatihan:

    - Proses pelatihan model dilakukan menggunakan platform Google Colaboratory dengan akselerator GPU NVIDIA Tesla T4 (15 GB VRAM) untuk mempercepat komputasi deep learning. Framework yang digunakan adalah PyTorch 2.5.1+cu121 dengan pustaka Ultralytics YOLOv11 (build 8.3.16) sebagai implementasi utama model YOLO [4],

Gambar 4 Lingkungan Google Colaboratory

- Hyperparameter Pelatihan:
  Untuk memastikan perbandingan yang adil antar skenario, hyperparameter pelatihan utama diseragamkan untuk semua model. Ini termasuk jumlah epoch (iterasi pelatihan melalui seluruh dataset) yang ditetapkan pada 100, ukuran batch (batch size, jumlah sampel yang diproses dalam satu iterasi) sebesar 16, dan ukuran citra input (imgsz) 640x640 piksel. Konsistensi hyperparameter krusial saat melakukan analisis komparatif [6]. Untuk menjaga validitas perbandingan antar model, seluruh percobaan menggunakan konfigurasi hyperparameter yang sama. Parameter pelatihan ditetapkan sebagai berikut: jumlah epoch sebanyak 100, batch size 16, dan ukuran citra input 640×640 piksel sesuai pada Gambar 5.

Gambar 5 Setup Hyperparameter

  Pemilihan parameter ini mengikuti praktik umum pada pelatihan model YOLO [6]. Nilai learning rate awal diatur pada 0.001 untuk optimizer SGD, dan 0.001 untuk Adam serta AdamW, dengan skema cosine learning rate scheduler untuk menjaga stabilitas konvergensi selama pelatihan. Semua eksperimen dijalankan dengan random seed tetap (42) untuk menjamin konsistensi hasil dan mencegah variance antar percobaan [12].

- Metrik Evaluasi: Kinerja setiap model yang telah dilatih dievaluasi pada test set (data yang tidak pernah dilihat model selama pelatihan). Tiga metrik utama digunakan untuk analisis cost-benefit:

DOI: https://doi.org/10.31004/riggs.v4i4.3329

- Akurasi Deteksi (Benefit): Diukur menggunakan mean Average Precision (mAP) pada ambang batas Intersection over Union (IoU) 0.5, disingkat mAP50. mAP adalah metrik standar untuk mengevaluasi model deteksi objek, yang mempertimbangkan presisi dan recall pada berbagai tingkat kepercayaan [9]. IoU 0.5 berarti deteksi dianggap benar jika area tumpang tindih antara kotak prediksi dan kotak ground truth minimal 50%. mAP50-95, yang menghitung rata-rata mAP pada ambang IoU dari 0.5 hingga 0.95 dengan interval 0.05, juga dicatat untuk memberikan gambaran kinerja yang lebih komprehensif.

- Kecepatan Inferensi (Benefit): Diukur dalam Frames Per Second (FPS) [7].

- Ukuran Model (Cost): Diukur dalam Megabytes (MB), merepresentasikan kebutuhan penyimpanan model dan seringkali berkorelasi dengan kompleksitas komputasi. Selain metrik utama ini, Precision (proporsi deteksi benar dari total deteksi) dan Recall (proporsi objek yang berhasil dideteksi dari total objek ground truth) juga dicatat, karena metrik ini umum digunakan dalam evaluasi klasifikasi dan deteksi [13, 10].

- Analisis Cost-Benefit:
  Untuk menentukan konfigurasi model yang paling optimal, dilakukan analisis kuantitatif berbasis cost–benefit dengan formula pembobotan pada persamaan (1):

$$Skor\ Akhir = (Norm_{mAP50}\ x\ 0.5) + (Norm_{FPS}\ x\ 0.3) + (Norm_{Model}\ x\ 0.2) \quad (1)$$

  Nilai benefit metrics (mAP50, FPS) dinormalisasi dalam rentang 0–1, sedangkan cost metric (ukuran model) dinormalisasi secara terbalik — semakin kecil ukuran, semakin tinggi skor.
  Bobot 50%–30%–20% diadaptasi dari studi benchmarking object detector efficiency yang mempertimbangkan keseimbangan antara akurasi dan efisiensi inferensi [14]. Pendekatan ini memungkinkan evaluasi obyektif terhadap performa YOLOv11 dalam konteks penerapan nyata seperti edge device deployment di bidang pertanian.

3. Hasil dan Diskusi

Pelaksanaan eksperimen dilakukan dalam lingkungan Google Colaboratory dengan GPU NVIDIA Tesla T4, menggunakan PyTorch dan framework Ultralytics. Evaluasi akhir dilakukan pada test set yang terdiri dari 24 citra yang belum pernah dilihat oleh model selama pelatihan.

3.1. Hasil Pengujian Kinerja Model

Hasil kuantitatif dari evaluasi sembilan kombinasi model disajikan dalam Tabel 1 (metrik akurasi) dan Tabel 2 (metrik efisiensi komputasi).

Tabel 1. Hasil Pengujian Akurasi Model

Model	Optimizer	Precision	Recall	F1-Score	mAP50	mAP50-95
yolo11m	SGD	0.911	0.902	0.906	0.948	0.746
yolo11m	Adam	0.851	0.818	0.834	0.878	0.637
yolo11m	AdamW	0.81	0.808	0.809	0.868	0.631
yolo11n	SGD	0.891	0.864	0.877	0.926	0.646
yolo11n	AdamW	0.806	0.761	0.783	0.853	0.608
yolo11n	Adam	0.785	0.791	0.788	0.839	0.602
yolo11s	SGD	0.884	0.885	0.884	0.933	0.696
yolo11s	Adam	0.824	0.789	0.806	0.88	0.629
yolo11s	AdamW	0.824	0.811	0.817	0.882	0.629

DOI: https://doi.org/10.31004/riggs.v4i4.3329

Tabel 2. Hasil Pengujian Efisiensi Komputasi

Model	Optimizer	Kecepatan Inferensi (CPU, FPS)	Ukuran Model (MB)
yolo11n	AdamW	4.62	5.5
yolo11n	Adam	3.54	5.5
yolo11n	SGD	3.49	5.5
yolo11s	SGD	1.74	19.2
yolo11s	AdamW	1.67	19.2
yolo11s	Adam	1.58	19.2
yolo11m	Adam	0.64	40.5
yolo11m	SGD	0.61	40.5
yolo11m	AdamW	0.6	40.5

Tabel 1 menunjukkan bahwa semua model mencapai tingkat mAP50 yang relatif tinggi (di atas 0.83), mengindikasikan kemampuan deteksi yang baik secara umum pada dataset ini. Namun, terdapat variasi yang cukup signifikan tergantung pada kombinasi arsitektur dan optimizer. Tabel 2 menyoroti perbedaan besar dalam hal efisiensi, di mana model 'n' jauh lebih cepat dan lebih kecil dibandingkan model 's' dan 'm'. Analisis Cost-Benefit dan Pemilihan Model Final Hasil analisis cost-benefit (dirangkum pada Tabel 3) secara objektif menentukan model mana yang memberikan keseimbangan performa terbaik.

3.2. Interpretasi Kinerja Optimizer

Analisis mendalam pada Tabel 1 mengungkapkan temuan yang signifikan mengenai peran optimizer. Optimizer SGD secara konsisten menunjukkan keunggulan dalam hal akurasi (mAP50) dibandingkan Adam dan AdamW untuk semua varian arsitektur (n, s, dan m) pada dataset spesifik ini. Model yolo11m dengan SGD mencapai mAP50 tertinggi (0.948), diikuti oleh yolo11s-SGD (0.933) dan yolo11n-SGD (0.926). Sebaliknya, model yang dilatih dengan Adam dan AdamW secara konsisten memiliki mAP50 yang lebih rendah untuk arsitektur yang sama.

Hasil ini memperkuat bahwa SGD memiliki stabilitas pelatihan yang lebih baik pada dataset berukuran kecil (254 citra). Optimizer ini menjaga gradien tetap konsisten antar-iterasi sehingga mengurangi risiko konvergensi prematur dan overfitting. Temuan ini sejalan dengan penjelasan Goodfellow et al. [5] bahwa adaptive optimizers seperti Adam dan AdamW dapat mempercepat konvergensi awal tetapi cenderung menyesuaikan bobot secara berlebihan pada low-data regime. Hal tersebut menyebabkan model terlalu menyesuaikan diri terhadap pola lokal dari data pelatihan dan kehilangan kemampuan generalisasi. Dengan demikian, hasil penelitian ini mendukung pendekatan konservatif berbasis SGD dalam konteks dataset terbatas, karena menawarkan keseimbangan antara stabilitas pelatihan dan kemampuan generalisasi model. Perlu dicatat bahwa superioritas optimizer dapat bergantung pada banyak faktor, termasuk ukuran dataset, kompleksitas masalah, arsitektur jaringan, dan tuning hyperparameter lainnya, sebagaimana juga terindikasi dalam studi analisis hyperparameter pada klasifikasi daging [4].

3.3. Analisis *Cost-Benefit* dan Pemilihan Model Final

Untuk mendapatkan kesimpulan yang objektif mengenai model mana yang paling sesuai untuk aplikasi praktis, analisis cost-benefit dilakukan dengan menggunakan skor terbobot yang telah dinormalisasi. Hasilnya dirangkum dalam Tabel 3.

Tabel 3. Hasil Analisis Cost-Benefit

Model	Optimizer	Norm. mAP50 (Benefit)	Norm. FPS (Benefit)	Norm. Ukuran (Cost)	Skor Akhir	Peringkat
yolo11n	SGD	0.798	0.72	1	0.839	1
yolo11n	AdamW	0.128	1	1	0.709	2
yolo11n	Adam	0	0.732	1	0.577	3
yolo11s	SGD	0.862	0.284	0.609	0.585	4
yolo11s	AdamW	0.395	0.266	0.609	0.423	5
yolo11s	Adam	0.376	0.244	0.609	0.41	6
yolo11m	SGD	1	0.002	0	0.334	7
yolo11m	Adam	0.358	0.01	0	0.123	8
yolo11m	AdamW	0.266	0	0	0.089	9

DOI: https://doi.org/10.31004/riggs.v4i4.3329

Berdasarkan skor akhir terbobot, model yolo11n - SGD muncul sebagai konfigurasi paling optimal dengan skor tertinggi (0.839). Analisis ini menyoroti pentingnya mempertimbangkan trade-off. Meskipun model yolo11m - SGD mencapai akurasi mAP50 tertinggi secara absolut (nilai ternormalisasi 1.000), kinerjanya dalam hal efisiensi sangat buruk. Kecepatan inferensinya sangat lambat (Norm. FPS ≈ 0.002) dan ukurannya paling besar (Norm. Ukuran = 0), menghasilkan skor akhir yang rendah (0.334) dan peringkat ke-7.

Sebaliknya, yolo11n - SGD menawarkan keseimbangan yang jauh lebih unggul. Meskipun akurasinya sedikit lebih rendah (Norm. mAP50 = 0.798) dibandingkan varian 's' dan 'm' dengan SGD, efisiensi komputasinya luar biasa. Model ini memiliki ukuran terkecil (5.5 MB, Norm. Ukuran = 1.00) dan kecepatan inferensi yang relatif baik (Norm. FPS = 0.72). Kombinasi antara akurasi yang masih sangat baik (mAP50 0.926) dengan efisiensi sumber daya yang ekstrem menjadikannya pilihan ideal untuk aplikasi praktis di lapangan, terutama jika target implementasinya adalah perangkat dengan keterbatasan komputasi (misalnya, perangkat seluler atau embedded systems) [9, 10]. Temuan ini sejalan dengan penelitian lain yang menekankan pentingnya keseimbangan kinerja dan ukuran model, seperti perbandingan arsitektur MobileNet, Inception ResNet V2, dan EfficientNet B2 [10].

Secara keseluruhan, hasil ini menegaskan bahwa YOLOv11n–SGD merupakan konfigurasi paling efisien secara komputasi. Meskipun tidak mencapai akurasi absolut tertinggi, model ini memiliki rasio efisiensi mAP50/FPS sebesar 0.265, tertinggi di antara seluruh kombinasi. Hal ini menunjukkan bahwa model ringan dengan performa stabil dapat memberikan keseimbangan optimal antara akurasi dan kecepatan, menjadikannya kandidat ideal untuk implementasi real-time inference pada perangkat edge computing seperti drone pertanian atau sistem monitoring berbasis IoT. Temuan ini selaras dengan prinsip Pareto efficiency dalam pemodelan deteksi objek yang menekankan optimasi kinerja tanpa mengorbankan efisiensi [12, 15]

Berdasarkan hasil tersebut, model YOLOv11n–SGD dipilih sebagai konfigurasi akhir yang digunakan dalam tahap kesimpulan dan rekomendasi penerapan sistem estimasi buah kelapa.

4. Kesimpulan

Penelitian ini mengusulkan pendekatan object detection berbasis YOLOv11 untuk estimasi jumlah buah kelapa pada pohon dengan kondisi latar belakang kompleks. Evaluasi dilakukan terhadap tiga varian arsitektur (YOLOv11n, YOLOv11s, dan YOLOv11m) serta tiga jenis optimizer (SGD, Adam, dan AdamW) menggunakan dataset berukuran kecil yang terdiri atas 254 citra beranotasi. Hasil pengujian menunjukkan bahwa optimizer SGD memberikan stabilitas pelatihan terbaik dalam konteks low-data regime, mendukung temuan sebelumnya oleh Goodfellow et al. [5]. Model YOLOv11n–SGD terbukti memberikan keseimbangan paling efisien dengan mAP50 sebesar 0.926, ukuran model 5.5 MB, dan rasio mAP50/FPS = 0.265, menjadikannya konfigurasi paling ideal untuk aplikasi real-time inference di perangkat dengan keterbatasan komputasi seperti drone pertanian dan edge devices. Kontribusi utama penelitian ini adalah penerapan kerangka analisis cost–benefit kuantitatif yang menggabungkan metrik akurasi, kecepatan inferensi, dan ukuran model sebagai dasar pemilihan konfigurasi terbaik. Pendekatan ini dapat diadaptasi untuk optimasi model deteksi objek lain yang beroperasi dalam kondisi sumber daya terbatas. Untuk penelitian selanjutnya, pengembangan diarahkan pada implementasi model YOLOv11n–SGD pada perangkat edge computing (misalnya Jetson Nano atau smartphone berbasis Android), serta perluasan sistem menuju deteksi berbasis segmentasi instance guna menangani oklusi antar buah pada pohon. Selain itu, penambahan dataset dengan variasi pencahayaan dan sudut pengambilan citra akan memperkuat kemampuan generalisasi model di lingkungan nyata.

Referensi

[1] N. K. A. Kristini Putri, I. A. Sintha Agustina, and N. L. Sintya Dewi, "Makna Filosofi Buah Kelapa Dalam Upakara Yadnya," *Majalah Ilmiah Untab*, vol. 19, no. 2, pp. 221–225, 2022.

[2] H. Gunawan and A. Asrof Fitri, "Praktik Jual Beli Padi Dengan Sistem Tebas Dan Ijon Melalui Perantara Dalam Perspektif Hukum Positif Dan Hukum Islam," *Penelitian Multidisiplin Ilmu*, vol. 1, no. 3, pp. 463–474, 2022, [Online]. Available: http://melatijournal.com/index.php/Metta

[3] S. Muhammad and S. Sambas, "Praktik Jual Beli Buah Petai Di Pohon Dengan Sistem Borongan Perspektif Fikih Muamalah," *Cross-border*, vol. 5, no. 2, pp. 1313–1321, 2022.

[4] A. Dinata, M. Sunarya, and G. Gunadi, "Analisis Hyperparameter Pada Klasifikasi Jenis Daging Menggunakan Algoritma Convolutional Neural Network I Made Anom Mahartha Dinata 1 , I Gede Aris Gunadi 2 , I Made Gede Sunarya," 2024.

[5] A. C. Ian Goodfellow, Yoshua Bengio, "Deep Learning," *MIT Press*, vol. 521, no. 7553, p. 785, 2017, doi: 10.1016/B978-0-12-391420-0.09987-X.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

DOI: https://doi.org/10.31004/riggs.v4i4.3329

[7] I Gusti Ngurah Bagus Putra Asmara, Made Windu Antara Kesiman, and Gede Indrawan, "Balinese Shadow Puppet Characters Detection In The Wayang Peteng Performance Using The Yolov5 Algorithm," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 12, no. 3, pp. 388–397, Dec. 2023, doi: 10.23887/janapati.v12i3.65906.

[8] N. Putu *et al.*, "TPHerbleaf : Dataset Untuk Klasifikasi Jenis Daun Tumbuhan Herbal Berdasarkan Lontar Usada Taru Pramana", [Online]. Available: https://s.id/jurnalresistor

[9] I. M. G. Sunarya, I Wayan Treman, and Putu Zasya Eka Satya Nugraha, "Classification of Rice Growth Stage on UAV Image Based on Convolutional Neural Network Method," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 12, no. 1, pp. 146–155, May 2023, doi: 10.23887/janapati.v12i1.60959.

[10] I. Made, A. Darma Putra, M. Dendi Maysanjaya, M. Windu, and A. Kesiman, "PENDEKATAN BERBASIS U-NET UNTUK SEGMENTASI HARD EXUDATE DALAM CITRA FUNDUS RETINA," *INSERT: Information System and Emerging Technology Journal*, vol. 4, no. 1, 2023.

[11] N. L. P. Kurniawati, M. W. Antara Kesiman, and I. M. G. Sunarya, "Recognition of Balinese Traditional Ornament Carving Images with Convolutional Neural Network and Discrete Wavelet Transform," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 8, no. 4, p. 670, Jan. 2023, doi: 10.26555/jiteki.v8i3.24360.

[12] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," Oct. 2024, [Online]. Available: http://arxiv.org/abs/2410.17725

[13] N. Putu Dita Ariani Sukma Dewi, M. Windu Antara Kesiman, I. Made Gede Sunarya, I. Gusti Ayu Agung Diatri Indradewi, and I. Gede Andika, "Classification of Herbal Plant Leaf Types Based on Lontar Usada Taru Pramana Using CNN," 2025.

[14] M. Yaseen, "What is YOLOv9: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector," Sep. 2024, [Online]. Available: http://arxiv.org/abs/2409.07813

[15] M. Kotthapalli, D. Ravipati, and R. Bhatia, "YOLOv1 to YOLOv11: A Comprehensive Survey of Real-Time Object Detection Innovations and Challenges," Aug. 2025, [Online]. Available: http://arxiv.org/abs/2508.02067

DOI: https://doi.org/10.31004/riggs.v4i4.3329