

Studi Perbandingan Model Hybrid MobileNet-YOLO: Trade-off Akurasi dan Efisiensi Komputasi dalam Deteksi Objek

Jayantiara Zalba Yuswa¹, Hotma Pangaribuan²

¹Mahasiswa Program Studi Teknik Informatika, Universitas Putera Batam

²Dosen Program Studi Teknik Informatika, Universitas Putera Batam

email: pb220210129@upbatam.ac.id

ABSTRACT

Deploying object detection models on resource-constrained devices requires balancing high accuracy with computational efficiency. This research analyzes the performance trade-off of hybrid architectures combining MobileNet backbones (V2, V3-Small, and V3-Large) with YOLO detectors. The study employs an experimental method using the Pascal VOC dataset across four input resolutions ranging from 320 to 640 pixels to measure Mean Average Precision (mAP@50) and Frames Per Second (FPS). The results demonstrate a significant trade-off between precision and speed. MobileNetV3-Large at 640px achieves the highest accuracy of 58.5% mAP, making it suitable for precision-critical tasks. Conversely, MobileNetV3-Small at 320px proves to be the most efficient model, achieving 165.2 FPS on a GPU and maintaining real-time performance on a standard CPU at 30.4 FPS. These findings provide empirical guidance for selecting optimal hybrid architectures based on specific hardware constraints, proving that MobileNetV3-Small is the superior choice for low-power edge implementation.

Keywords: *Computational Efficiency; Hybrid Model; MobileNet; Object Detection; YOLO.*

PENDAHULUAN

Perkembangan teknologi *Computer Vision* dalam satu dekade terakhir telah mendorong implementasi sistem cerdas pada berbagai sektor strategis, mulai dari pemantauan keamanan hingga transportasi cerdas (Ramdan et al., 2024). Namun, penerapan teknologi ini pada perangkat *edge* dan *Internet of Things* (IoT) masih menghadapi kendala serius akibat keterbatasan sumber daya komputasi dan memori. Algoritma deteksi objek konvensional seperti YOLO standar, meskipun memiliki akurasi tinggi, sering kali membebani prosesor kecil sehingga

menyebabkan latensi tinggi yang menurunkan responsivitas sistem. Oleh karena itu, diperlukan strategi optimasi melalui pengembangan arsitektur hibrida yang mampu menyeimbangkan ketelitian deteksi tinggi dengan efisiensi daya komputasi (Su et al., 2025).

Penelitian terdahulu telah berupaya memodifikasi arsitektur *Convolutional Neural Network* (CNN) menjadi lebih ringkas. Tinjauan literatur sistematis oleh Wang et al., 2025, menunjukkan bahwa tren pengembangan model *lightweight* terus meningkat tajam guna mendukung implementasi AI pada perangkat terbatas. Senada dengan hal tersebut, Jiang et al.,

2025, mengemukakan bahwa penggunaan *backbone* ringan seperti MobileNet dapat mereduksi ukuran model secara signifikan tanpa mengorbankan akurasi secara fatal. Sementara itu, Terven et al., 2023, menyoroti adaptabilitas evolusi algoritma YOLO terhadap berbagai jenis perangkat keras.

Meskipun demikian, literatur saat ini belum secara komprehensif memetakan perbandingan *head-to-head* antara generasi MobileNetV2 dan MobileNetV3 (varian *Small* dan *Large*) sebagai pengekstraksi fitur pada detektor YOLO. Ketiadaan data empiris yang mendalam mengenai pengaruh resolusi input terhadap *trade-off* akurasi dan kecepatan menjadi celah penelitian krusial yang perlu dijawab.

Penelitian ini mengusulkan pengembangan model hibrida yang mengintegrasikan efisiensi MobileNet sebagai *feature extractor* dan kecepatan YOLO sebagai *detection head*. Pendekatan ini didasarkan pada keunggulan blok *Squeeze-and-Excitation* pada MobileNetV3 yang mampu meningkatkan representasi fitur spasial tanpa menambah beban komputasi berlebih. Tujuan utamanya adalah menganalisis kinerja MobileNetV2-YOLO, MobileNetV3-Small-YOLO, dan MobileNetV3-Large-YOLO pada empat tingkat resolusi input (320, 416, 512, dan 640 piksel). Evaluasi dilakukan menggunakan metrik *Mean Average Precision (mAP@50)* dan *Frames Per Second (FPS)* untuk membuktikan hipotesis bahwa varian *Small* unggul dalam kecepatan *real-time*, sedangkan varian *Large* menawarkan presisi lebih tinggi.

Melalui eksperimen komprehensif menggunakan dataset Pascal VOC, penelitian ini menghasilkan panduan

empiris mengenai titik keseimbangan (*trade-off*) terbaik untuk spesifikasi perangkat keras berbasis GPU maupun CPU. Kontribusi utama penelitian ini adalah pemetaan karakteristik performa yang membuktikan bahwa resolusi input merupakan variabel kontrol yang sama pentingnya dengan pemilihan arsitektur. Hasil ini memberikan rekomendasi teknis konkret bagi implementasi sistem pengawasan cerdas dan robotika pada perangkat terbatas di masa depan.

KAJIAN TEORI

2.1 Object Detection

Object detection merupakan bagian dari computer vision yang bertujuan mengidentifikasi keberadaan objek serta menentukan lokasi objek pada suatu gambar atau video melalui bounding box. Metode modern berbasis deep learning mengintegrasikan proses klasifikasi dan regresi dalam satu sistem prediksi. Pendekatan *one-stage detector* seperti YOLO lebih efisien dibanding *two-stage detector* karena inferensi hanya dilakukan satu kali (*single forward pass*), sehingga mendukung penerapan real-time terutama pada perangkat dengan keterbatasan sumber daya komputasi (Ashari et al., 2025).

2.2 YOLO (You Only Look Once)

YOLO merupakan algoritma deteksi objek satu tahap yang mengubah permasalahan deteksi menjadi regresi langsung terhadap koordinat objek dan kategorinya. Varian YOLO ringan seperti YOLOv4-Tiny dan YOLOv4-Lite dirancang untuk mengurangi kompleksitas komputasi dan mempercepat inferensi tanpa menurunkan akurasi secara drastis. Model-model YOLO banyak diterapkan pada sistem pengawasan dan deteksi

real-time karena keunggulan kecepatannya (Jiang et al., 2025).

2.3 MobileNet

MobileNet merupakan arsitektur CNN yang dikembangkan untuk perangkat bersumber daya rendah melalui penggunaan *depthwise separable convolution*, yaitu pemisahan proses konvolusi yang dapat menurunkan jumlah parameter dan *computational cost* secara signifikan. Integrasi MobileNet sebagai backbone pada sistem lightweight object detection bertujuan mempercepat inferensi dan menekan ukuran model, sekaligus tetap menjaga performa deteksi kompetitif (Jiang et al., 2025).

2.4 Hybrid MobileNet–YOLO

Hybrid MobileNet–YOLO menggabungkan backbone MobileNet dengan head deteksi YOLO untuk mencapai efisiensi komputasi yang tinggi tanpa menurunkan akurasi secara ekstrem. Studi MOLO membuktikan bahwa penggantian backbone YOLO dengan MobileNetV2 mampu menurunkan konsumsi memori dan meningkatkan kecepatan inferensi pada perangkat edge seperti IoT device, sambil mempertahankan akurasi pada kategori acceptable (Khurana et al., 2025).

2.5 Trade-Off Akurasi dan Efisiensi Komputasi

Reduksi kompleksitas arsitektur model umumnya berkonsekuensi pada turunnya akurasi, terutama terkait objek kecil dan kompleksitas lingkungan. Pada penelitian YOLOv4-Lite, proses *model lightweighting* terbukti meningkatkan FPS namun menurunkan mAP secara moderat dalam skenario real-time (Jiang et al., 2025). Sementara itu, studi MOLO menunjukkan bahwa pemilihan backbone

yang tepat dapat menekan dampak penurunan akurasi meskipun parameter model dipangkas secara signifikan (Khurana et al., 2025).

2.6 Penelitian Terdahulu dan Posisi Penelitian Ini (State of The Art)

Penelitian Ashari et al., 2025, berfokus pada lightweight YOLO untuk multi-vehicle detection, namun tidak membahas perubahan backbone. Studi Mobile-YOLO Jiang et al., 2025, menggunakan backbone ringan namun tidak menguji beberapa varian MobileNet dalam satu eksperimen. MOLO Khurana et al., 2025 mengintegrasikan MobileNetV2, tetapi tidak melakukan komparasi antar model maupun evaluasi resolusi input secara variatif.

Penelitian serupa juga dilakukan oleh (Nugroho & Baihaqi, 2023) yang mengganti *backbone* YOLOv5 dengan MobileNetV3s, namun fokus studi tersebut terbatas pada deteksi atribut sekolah dan tidak membandingkan performa lintas generasi MobileNet (V2 vs V3).

Penelitian ini memiliki posisi berbeda karena menguji tiga konfigurasi hybrid secara komparatif:

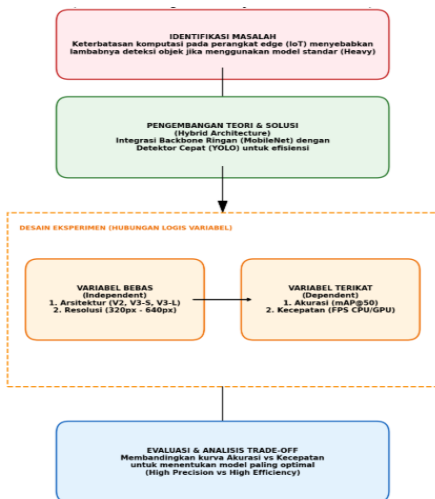
1. MobileNetV3-Small + YOLOv4-Tiny
2. MobileNetV2 + YOLOv4-Tiny
3. MobileNetV3-Large + YOLOv5-Nano

serta melakukan evaluasi multi-metrik meliputi akurasi (mAP), kompleksitas model (jumlah parameter & ukuran file), dan waktu inferensi pada beberapa variasi ukuran input (misalnya 640 px dan ukuran lain sesuai hasil eksperimen). Kebaruan ini memberikan gambaran komprehensif mengenai batas efisiensi dan akurasi untuk pendekatan hybrid backbone ringan

dalam konteks deteksi objek berbasis deep learning.

2.7 Kerangka Pemikiran

Berikut merupakan kerangka pemikiran ini dirancang secara sistematis untuk memandu jalannya eksperimen mulai dari tahap awal hingga penarikan kesimpulan.

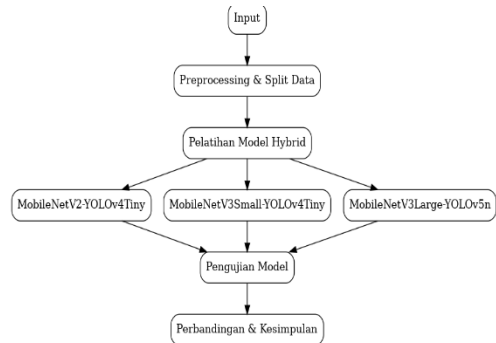


Gambar 1 Kerangka Pemikiran (Sumber: Data Penelitian, 2025)

METODE PENELITIAN

3.1. Tahapan Penelitian

Penelitian ini menggunakan metode eksperimental kuantitatif untuk mengukur dan membandingkan kinerja model deteksi objek. Tahapan penelitian dirancang secara sistematis dimulai dari identifikasi masalah, pengumpulan data, pra-pemrosesan data, perancangan model, pelatihan (*training*), hingga evaluasi dan analisis hasil. Seluruh tahapan proses pelaksanaan penelitian digambarkan dalam bagan alir (*flowchart*) berikut ini:



Gambar 2 Kerangka Operasional (Sumber: Data Penelitian, 2025)

Secara garis besar, alur penelitian terdiri dari enam tahapan utama sebagai berikut:

1. Input (Masukan Data) Tahap ini merupakan langkah inisiasi penelitian yang melibatkan pengumpulan dataset citra objek. Dalam konteks penelitian ini, input merujuk pada dataset standar (seperti Pascal VOC) yang akan digunakan sebagai bahan utama pembelajaran model.
2. Preprocessing & Split Data Sebelum masuk ke tahap pelatihan, data mentah melalui proses pra-pemrosesan (*preprocessing*). Proses ini mencakup penyesuaian ukuran citra (*resizing*) ke dalam variasi resolusi (misalnya 320px, 416px, 512px, 640px) dan normalisasi piksel. Selanjutnya, data dibagi (*split data*) menjadi tiga subset utama: data latih (*training set*) untuk melatih model, data validasi (*validation set*) untuk memantau proses belajar, dan data uji (*testing set*) untuk evaluasi akhir.
3. Pelatihan Model Hybrid Tahap inti penelitian adalah pelatihan model *hybrid*. Penelitian ini menerapkan teknik *Transfer Learning* (Zhuang et

al., 2021) dengan menggabungkan arsitektur *backbone* (jaringan pengekstraksi fitur) yang ringan dengan *head* (jaringan detektor) yang cepat. Pelatihan dilakukan secara terpisah pada tiga variasi arsitektur yang berbeda untuk melihat pengaruh kompleksitas model:

MobileNetV2-YOLOv4Tiny: Menguji kombinasi arsitektur MobileNet generasi kedua dengan detektor YOLOv4 versi ringan.

MobileNetV3Small-YOLOv4Tiny: Menguji varian arsitektur MobileNetV3 yang paling efisien (Small) untuk mengukur kecepatan maksimal.

MobileNetV3Large-YOLOv5n: Menguji varian MobileNetV3 yang lebih kompleks (Large) dengan YOLOv5 Nano untuk mengejar akurasi yang lebih tinggi.

4. Pengujian Model (Testing) Setelah model selesai dilatih (*trained models*), dilakukan pengujian menggunakan data uji yang belum pernah dilihat model sebelumnya. Pengujian ini bertujuan untuk mengukur performa model secara objektif menggunakan parameter kuantitatif, yaitu akurasi deteksi (*Mean Average Precision / mAP*) dan kecepatan inferensi (*Frames Per Second / FPS*).
5. Perbandingan & Kesimpulan Tahap akhir adalah melakukan analisis komparatif terhadap hasil pengujian ketiga model tersebut. Data mAP dan FPS dibandingkan untuk menemukan *trade-off* atau keseimbangan terbaik antara akurasi dan kecepatan. Berdasarkan analisis tersebut, ditarik kesimpulan mengenai model mana yang paling

optimal untuk diimplementasikan pada perangkat dengan sumber daya terbatas.

3.2. Lokasi dan Waktu Penelitian

Penelitian ini dilaksanakan secara daring menggunakan *platform cloud computing* Kaggle dan Google Colab untuk memfasilitasi proses pelatihan model yang membutuhkan sumber daya komputasi GPU (*Graphics Processing Unit*) tingkat tinggi. Waktu penelitian dilaksanakan mulai bulan September 2025 hingga Desember 2025.

3.3. Variabel Penelitian

Variabel yang digunakan dalam penelitian ini terdiri dari variabel bebas (*independent variable*) dan variabel terikat (*dependent variable*):

1. Variabel Bebas: Arsitektur *Backbone* (MobileNetV2, MobileNetV3-Small, MobileNetV3-Large) dan Resolusi Input Citra (320x320, 416x416, 512x512, 640x640 piksel).
2. Variabel Terikat: Akurasi deteksi yang diukur dengan *Mean Average Precision* (mAP@50) dan kecepatan inferensi yang diukur dengan *Frames Per Second* (FPS).

3.4. Populasi dan Sampel Penelitian

Populasi dalam penelitian ini adalah citra digital yang memuat objek-objek umum. Sampel yang digunakan adalah dataset standar Pascal VOC 2012 (*Visual Object Classes*). Teknik pengambilan sampel dilakukan dengan metode *purposive sampling*, yaitu memilih dataset yang telah terstandarisasi untuk *benchmark* deteksi objek. Dataset dibagi menggunakan teknik *hold-out validation* dengan proporsi data latih (*training*), data validasi, dan data uji (*testing*) untuk

memastikan model dievaluasi secara objektif.

3.5. Hipotesis Penelitian

Berdasarkan perumusan masalah, hipotesis yang diajukan dalam penelitian ini adalah:

1. H1: Peningkatan resolusi input citra akan meningkatkan nilai mAP namun menurunkan nilai FPS secara signifikan.
2. H2: Model dengan *backbone* MobileNetV3-Large akan menghasilkan akurasi tertinggi dibandingkan varian lainnya.
3. H3: Model dengan *backbone* MobileNetV3-Small akan menghasilkan kecepatan inferensi (FPS) tertinggi dan paling efisien untuk perangkat terbatas.

3.6. Model dan Rancangan Penelitian

Rancangan penelitian yang digunakan adalah eksperimen faktorial di mana setiap kombinasi *backbone* dan resolusi diujikan. Model penelitian yang dibangun adalah arsitektur *Hybrid* yang menggabungkan MobileNet (sebagai pengekstraksi fitur) dan algoritma YOLO (sebagai detektor). Model ini dirancang untuk meminimalkan beban komputasi (FLOPs) tanpa mengurangi kemampuan deteksi secara drastis, sehingga cocok untuk implementasi *mobile*.

3.7. Teknik Pengumpulan Data

Data dikumpulkan melalui metode dokumentasi digital dengan mengunduh dataset Pascal VOC dari repositori publik. Data citra dan label anotasi (*bounding box*) diverifikasi kelengkapannya sebelum masuk ke tahap pra-pemrosesan. Selain itu, data hasil eksperimen (nilai *loss*, mAP, presisi, *recall*, dan waktu inferensi) dikumpulkan secara otomatis melalui *log*

sistem selama proses pelatihan dan pengujian model berlangsung.

3.8. Teknik Analisis Data

Analisis data dilakukan secara deskriptif komparatif. Data hasil pengujian diolah untuk menghasilkan:

1. Analisis Akurasi: Membandingkan nilai mAP@50 antar model menggunakan tabel dan grafik batang.
2. Analisis Kecepatan: Membandingkan nilai FPS untuk menentukan kelayakan *real-time* (>30 FPS).
3. Analisis *Trade-off*: Menggunakan *scatter plot* untuk melihat hubungan antara kompleksitas model (parameter) dengan akurasi yang dihasilkan.

HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

Penelitian ini berfokus pada analisis *trade-off* antara akurasi deteksi dan efisiensi komputasi pada model hibrida MobileNet-YOLO. Eksperimen dilakukan terhadap tiga arsitektur (*MobileNetV2*, *MobileNetV3-Small*, *MobileNetV3-Large*) dengan empat variasi resolusi input citra (320, 416, 512, dan 640 piksel). Data kuantitatif hasil pengujian yang meliputi akurasi (*Mean Average Precision / mAP@50*), kecepatan inferensi (FPS), dan ukuran model dirangkum pada Tabel 1.

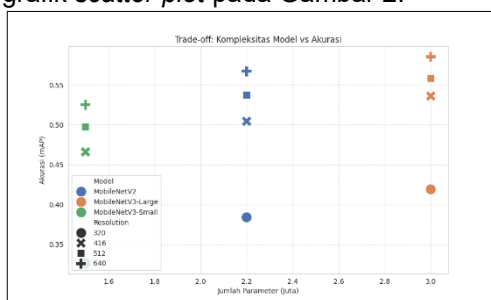
Table 1 Rekapitulasi Perbandingan Akurasi, Ukuran, dan Kecepatan Model

Model	Resolution	mAP50	File Size (MB)	FPS GPU	FPS CPU
MobileNetV2	320	38,39%	4.24	163.0	25.4
MobileNetV3-Large	320	41,90%	5.93	161.5	27.4
MobileNetV3-Small	320	32,64%	2.70	165.2	30.4
MobileNetV2	416	50,44%	4.25	156.5	17.0
MobileNetV3-Large	416	53,60%	5.94	158.4	19.7
MobileNetV3-Small	416	46,60%	2.70	162.1	22.0
MobileNetV2	512	53,71%	4.24	157.4	13.3
MobileNetV3-Large	512	55,83%	5.93	152.8	15.1
MobileNetV3-Small	512	49,72%	2.70	153.8	17.2
MobileNetV2	640	56,70%	4.25	142.5	9.5
MobileNetV3-Large	640	58,53%	5.94	142.4	9.6
MobileNetV3-Small	640	52,50%	2.71	142.3	12.3

(Sumber: Data Penelitian, 2025)

4.1.1. Analisis Trade-off: Kompleksitas dan Akurasi

Untuk memvalidasi judul penelitian mengenai "Trade-off", hubungan antara kompleksitas model (jumlah parameter) dan akurasi divisualisasikan dalam bentuk grafik *scatter plot* pada Gambar 2.



Gambar 3 Analisis Trade-off Antara Ukuran Model dan Akurasi (Sumber: Data Penelitian, 2025)

Berdasarkan Gambar 3, terlihat pola distribusi yang membentuk tiga kluster utama:

1. MobileNetV3-Large (Ikon Oranye): Berada di posisi paling kanan atas, menunjukkan model ini memiliki kompleksitas tertinggi (Parameter ~3.0 Juta) namun sebanding dengan hasil akurasi tertingginya (58.5% pada 640px). Ini adalah pilihan terbaik jika prioritas utama adalah ketelitian.
2. MobileNetV2 (Ikon Biru): Berada di posisi tengah (Parameter ~2.2 Juta). Model ini menawarkan keseimbangan moderat dengan akurasi mencapai 56.7% pada 640px. Posisi ini menunjukkan bahwa meskipun V2 adalah arsitektur yang lebih tua, kinerjanya masih kompetitif, namun mulai terkejar efisiensinya oleh optimasi arsitektur pada generasi V3.
3. MobileNetV3-Small (Ikon Hijau): Berada di posisi paling kiri bawah, menegaskan statusnya sebagai model paling ringan (Parameter ~1.5 Juta). Meskipun akurasinya paling rendah (52.5%), model ini memiliki "biaya" komputasi termurah, yang

sangat krusial untuk perangkat keras terbatas.

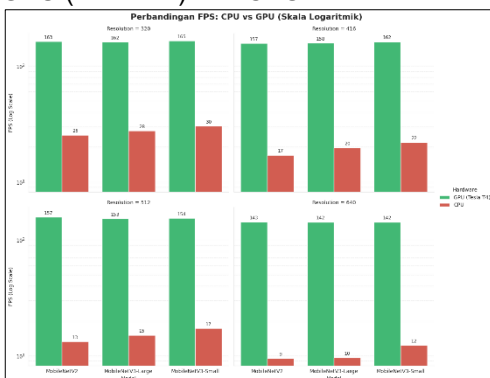
4. Pengaruh Resolusi: Secara vertikal, setiap model mengalami lonjakan akurasi yang signifikan ketika resolusi ditingkatkan dari 320px ke 640px, membuktikan bahwa dimensi input memiliki dampak langsung terhadap kemampuan model mengenali fitur objek tanpa menambah jumlah parameter (posisi horizontal tetap).

4.1.2. Analisis Efisiensi Komputasi (FPS)

Efisiensi komputasi diukur menggunakan metrik *Frames Per Second* (FPS). Perhitungan FPS didasarkan pada total waktu latensi dengan formula matematis sebagai berikut:

$$FPS = \frac{1000}{t_{preprocess} + t_{inference} + t_{nms}}$$

Keterangan: Variabel t merepresentasikan waktu proses dalam satuan milidetik (ms). Hasil benchmark pada Gambar 4 memperlihatkan perbedaan performa ekstrem antara penggunaan akselerator GPU (Tesla T4) dan CPU standar.



Gambar 4 Perbandingan Kecepatan Inferensi GPU vs CPU (Skala Logaritmik) (Sumber: Data Penelitian, 2025)

Data menunjukkan bahwa pada lingkungan GPU, seluruh model mampu berjalan sangat cepat (>140 FPS). Namun, pengujian pada CPU (yang merepresentasikan perangkat *edge* terbatas) menunjukkan hasil kritis: Hanya MobileNetV3-Small pada resolusi 320px yang mampu menembus ambang batas *real-time* 30 FPS (tepatnya 30.4 FPS). Model lainnya pada resolusi tinggi mengalami penurunan drastis hingga di bawah 10 FPS.

4.1.3. Hasil Deteksi Visual

Pengujian kualitatif dilakukan untuk membandingkan kemampuan deteksi objek secara visual. Gambar 5 menampilkan komparasi side-by-side pada resolusi tertinggi (640px).



Gambar 5 Perbandingan Visual Hasil Deteksi pada Resolusi 640px (Sumber: Data Penelitian, 2025)

Visualisasi pada Gambar 5, menunjukkan bahwa MobileNetV3-Large mampu mendeteksi objek dengan tingkat kepercayaan (*confidence score*) tertinggi, misalnya mendeteksi monitor dengan skor 0.72, dibandingkan MobileNetV2 (0.26) dan MobileNetV3-Small (0.60). Hal ini mengonfirmasi data kuantitatif bahwa varian Large memiliki presisi terbaik.

4.2. Pembahasan

4.2.1. Analisis Kritis Trade-off

Penelitian ini menemukan bahwa titik keseimbangan (*sweet spot*) sangat bergantung pada ketersediaan perangkat keras. Jika sistem menggunakan GPU,

MobileNetV3-Large (640px) adalah pilihan mutlak karena memberikan akurasi maksimal (58.5%) dengan kecepatan yang masih sangat tinggi (142.4 FPS). Namun, untuk implementasi pada perangkat tanpa GPU (CPU Only), MobileNetV3-Small (320px) adalah satu-satunya solusi yang layak karena merupakan satu-satunya konfigurasi yang mencapai performa *real-time* (30.4 FPS), meskipun harus mengorbankan akurasi menjadi 32.6%. Fenomena ini membuktikan bahwa efisiensi komputasi pada perangkat *edge* tidak hanya dipengaruhi oleh arsitektur model, tetapi juga sangat sensitif terhadap dimensi input.

4.2.2. Perbandingan dengan Penelitian Terdahulu (State of the Art)

Temuan ini memperkaya studi Jiang et al. (2022) yang membahas deteksi objek ringan, menekankan pada pemangkasan model, penelitian ini membuktikan bahwa manipulasi resolusi input pada MobileNetV3 memberikan kontrol yang lebih granular terhadap FPS dibandingkan sekadar mengganti arsitektur.

Temuan efisiensi pada MobileNetV3-Small ini selaras dengan hasil eksperimen Zhang et al., 2023, pada kasus deteksi hama, di mana integrasi MobileNetV3 sebagai *backbone* terbukti mampu mereduksi parameter model secara signifikan hingga 80% dibandingkan standar, tanpa penurunan akurasi yang fatal.

Selain itu, hasil ini sejalan dengan tinjauan Terven et al., 2023, mengenai evolusi YOLO, di mana integrasi *backbone* yang efisien seperti MobileNetV3 terbukti mampu menjaga stabilitas inferensi pada hardware terbatas, berbeda dengan *backbone* CSPDarknet standar yang cenderung berat. Hal ini juga dikuatkan oleh penelitian (Nugroho & Baihaqi, 2023)

yang menyimpulkan bahwa penggantian *backbone* ke MobileNetV3s memberikan peningkatan FPS yang krusial untuk aplikasi atribut sekolah secara *real-time*.

4.2.3. Kontribusi Keilmuan (Body of Knowledge)

Kontribusi utama penelitian ini adalah pemetaan empiris batas performa model hibrida. Penelitian ini menegaskan bahwa dalam perancangan sistem deteksi objek pada Internet of Things (IoT), pengembang tidak bisa hanya terpaku pada metrik akurasi (mAP). Variabel Resolusi Input harus diperlakukan sebagai hiperparameter kritis yang menentukan kelayakan implementasi sistem. Penelitian ini menyumbangkan rekomendasi teknis konkret: gunakan resolusi 320px untuk latensi rendah pada CPU, dan 640px untuk presisi tinggi pada GPU.

SIMPULAN

Berdasarkan hasil pengujian dan analisis yang telah dilakukan terhadap model hibrida MobileNetV2-YOLO, MobileNetV3-Small-YOLO, dan MobileNetV3-Large-YOLO pada berbagai resolusi input, dapat ditarik beberapa simpulan sebagai berikut:

1. Pengaruh Arsitektur dan Resolusi: Terdapat hubungan *trade-off* yang signifikan antara akurasi dan efisiensi komputasi. Peningkatan resolusi input dari 320 piksel ke 640 piksel secara konsisten meningkatkan akurasi deteksi (*Mean Average Precision*), namun menurunkan kecepatan inferensi (*FPS*) secara drastis, terutama pada lingkungan komputasi tanpa akselerator GPU.
2. Performa Akurasi Terbaik: Model MobileNetV3-Large dengan resolusi input 640 piksel terbukti menjadi

konfigurasi paling presisi dengan nilai mAP@50 mencapai 58.5%. Model ini paling direkomendasikan untuk skenario pengawasan yang mengutamakan ketelitian deteksi objek dan memiliki dukungan perangkat keras yang memadai (GPU).

3. Performa Efisiensi Terbaik: Model MobileNetV3-Small dengan resolusi input 320 piksel merupakan solusi paling efisien. Model ini mencatatkan kecepatan inferensi tertinggi mencapai 165.2 FPS pada GPU Tesla T4. Lebih penting lagi, model ini adalah satu-satunya varian yang mampu mencapai performa *real-time* pada pengujian menggunakan CPU standar, dengan kecepatan rata-rata 30.4 FPS.
4. Komparasi Antar Generasi: Arsitektur MobileNetV3 terbukti lebih unggul dibandingkan MobileNetV2 dalam hal fleksibilitas. Varian V3-Large mampu mengungguli akurasi V2 (58.5% vs 56.7%), sementara varian V3-Small mampu mengungguli kecepatan V2 secara signifikan pada perangkat terbatas. Hal ini menunjukkan bahwa penggunaan fungsi aktivasi *h-swish* dan modul *Squeeze-and-Excitation* pada MobileNetV3 memberikan dampak positif terhadap kinerja model hibrida YOLO.
5. Rekomendasi Implementasi: Pemilihan model harus disesuaikan dengan sumber daya perangkat keras. Untuk implementasi pada *server* atau *cloud* dengan GPU, disarankan menggunakan MobileNetV3-Large (640px). Sedangkan untuk implementasi pada perangkat *edge* atau *mobile* yang hanya mengandalkan CPU, MobileNetV3-Small (320px) adalah

pilihan mutlak untuk menjaga responsivitas sistem.

DAFTAR PUSTAKA

- Ashari, I., Negara, I. S. M., & A, A. S. S. (2025). Lightweight YOLO Models for Real-Time Multi-Vehicle Detection. *Sinkron*, 9(3), 1795–1810. <https://doi.org/10.33395/sinkron.v9i3.15071>
- Jiang, H., Zhao, J., Ma, F., Yang, Y., & Yi, R. (2025). Mobile-YOLO: A Lightweight Object Detection Algorithm for Four Categories of Aquatic Organisms. *Fishes*, 10(7). <https://doi.org/10.3390/fishes10070348>
- Khurana, K., Sonsare, P., Borkar, D., Thakkar, H., & Bhagat, O. (2025). MOLO: a hybrid approach using MobileNet and YOLO for object detection on resource constrained devices. *Discover Artificial Intelligence*, 5(1). <https://doi.org/10.1007/s44163-025-00398-3>
- Nugroho, A. D., & Baihaqi, W. M. (2023). Improved YOLOv5 with Backbone Replacement to MobileNet V3s for School Attribute Detection. *SinkrOn*, 8(3), 1944–1954. <https://doi.org/10.33395/sinkron.v8i3.12702>
- Ramdan, A., & Asriyanik, A. (2024). Implementasi Deteksi Objek Real-Time Sebagai Media Edukasi dengan Algoritma YOLOv8 pada Objek Sampah. *Jurnal SAINTEKOM*, 14(2), 142–153. <https://doi.org/10.33020/saintekom.v14i2.638>
- Su, C., Zhu, L., Dai, W., Zhou, J., Wang, J., Mao, Y., & Sun, J. (2025). Nav-YOLO: A Lightweight and Efficient

- Object Detection Model for Real-Time Indoor Navigation on Mobile Platforms. *ISPRS International Journal of Geo-Information*, 14(9). <https://doi.org/10.3390/ijgi14090364>
- Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. In *Machine Learning and Knowledge Extraction* (Vol. 5, Issue 4, pp. 1680–1716). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/make5040083>
- Wang, L., Wang, H., Letchmunan, S., Xiao, R., Ahmed, O. H., & Liu, Z. (2025). A systematic literature review of lightweight YOLO models for object detection. *PeerJ Computer Science*, 11, e3357. <https://doi.org/10.7717/peerj-cs.3357>
- Zhang, J., Wang, J., & Zhao, M. (2023). A Lightweight Crop Pest Detection Algorithm Based on Improved Yolov5s. *Agronomy*, 13(7). <https://doi.org/10.3390/agronomy13071779>
- Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. In *Proceedings of the IEEE* (Vol. 109, Issue 1, pp. 43–76). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/JPROC.2020.3004555>

	<p>Jayantiara Zalba Yuswa Merupakan Mahasiswa Prodi Teknik Informatika di Universitas Putera Batam.</p>
	<p>Hotma Pangaribuan Merupakan Dosen Prodi Teknik Informatik Universitas Putera Batam. Aktif sebagai tenaga kerja dan peneliti di bidangnya.</p>