

Eksplorasi Deteksi dan Segmentasi Objek Menggunakan Mask R-CNN Berbasis Detectron2 sebagai Dasar Prediksi Ukuran

Amadea Permana Sanusi^{1*}, Andriyan Rizki Jatmiko², Dian Fitri Islamiati Munisah³

^{1,2,3} Sistem Informasi, Fakultas Teknologi Informasi, Universitas Merdeka Malang, Indonesia

¹amadea@unmer.ac.id, ²andriyan.jatmiko@unmer.ac.id, ³dianmunisah1@gmail.com

Abstrak

Perkembangan teknologi pada zaman sekarang semakin berkembang dengan waktu cepat terutama pengembangan dalam bidang *computer vision*. Salah satu pengembangan yang dapat dikaji lebih dalam pada bidang ini adalah implementasi dari *instance segmentation*. Penelitian ini menggunakan dataset custom berisi 100 citra dengan dua kelas objek (*handphone* dan *flashdisk*) yang dianotasi menggunakan format COCO dengan rasio 70% *training* (70 citra), 25% *validation* (25 citra), dan 5% *testing* (5 citra). Pada penelitian ini merupakan eksplorasi awal dalam implementasi algoritma Mask R-CNN menggunakan *framework* Detectron2 dengan konfigurasi *learning rate* 0.02, maksimum iterasi 270.000, *batch size* 256, *backbone* ResNet-FPN, dan *threshold confidence* 0.5. Evaluasi menggunakan mAP dengan nilai parameter yang dihasilkan yaitu AP sebesar 62.615%, AP50 sebesar 71.836%, AP75 sebesar 67.884%, dan APl sebesar 62.615% yang mengindikasikan performa deteksi cukup baik. Namun, berdasarkan analisis dari hasil visual menunjukkan bahwa objek yang tumpang tindih masih menjadi tantangan bagi model serta *under-detection* menjadi permasalahan yang muncul selama proses pengujian. Peningkatan dapat dilakukan agar model dapat mendeteksi secara lebih akurat dengan melakukan beberapa hal seperti melakukan penyesuaian terhadap parameter *Non-Maximum Suppression* (NMS) dan ukuran *anchor* agar lebih sesuai dengan karakteristik objek target. Selain itu, penerapan teknik augmentasi data seperti *flipping* dan *occlusion* dapat memperkaya variasi dalam dataset. Penambahan citra yang memuat objek berukuran kecil hingga sedang juga penting untuk meningkatkan keberagaman data. Upaya perbaikan ini menjadi langkah penting dalam mendukung pengembangan penelitian pada tahap selanjutnya.

Kata kunci: Detectron2, Instance Segmentation, Mask R-CNN, Mean Average Precision (mAP)

1. Pendahuluan

Perkembangan teknologi pada zaman sekarang semakin berkembang dengan waktu cepat terutama pengembangan dalam bidang *computer vision*. Teknologi ini berfokus pada bagaimana sebuah komputer dapat melihat, memahami, maupun menginterpretasi objek dalam bentuk gambar ataupun video digital sebagaimana indra pengelihatan pada manusia (Hermansyah & Fadillah, 2019; Vidia et al., n.d.; Yunita & Setyati, 2019). *Computer vision* sendiri telah banyak diimplementasikan dalam berbagai bidang seperti pertanian, dunia industri, medis, maupun transportasi (Supiyandi Supiyandi & Rafif Rasendriya, 2024). Salah satu percabangan dalam *computer vision* adalah teknik *deep learning* dengan fitur segmentasinya yang sangat bermanfaat untuk membedakan objek satu dengan lainnya. Dalam penerapan segmentasi objek memiliki dua jenis segmentasi yaitu *instance segmentation* dan *semantic segmentation* (Rasyad, 2025). Penggunaan jenis segmentasi ini bergantung pada tujuan penerapan *deep learning*, sebagai contoh jika segmentasi objek membutuhkan hasil yang lebih detail maka *instance segmentation* bisa menjadi

solusi yang tepat. Tingkat akurasi dalam segmentasi menjadi hal penting dalam penerapan *deep learning* sebagai tujuan akhir pengembangan implementasi dan dasar pengambilan keputusan otomatis seperti pengukuran objek, klasifikasi lanjutan, dan lain sebagainya (Mahdiyah et al., 2025).

Berlandaskan kebutuhan industri akan sistem deteksi objek yang tidak hanya mampu mengenali keberadaan objek, tetapi juga memberikan informasi geometris presisi hingga tingkat piksel. Dalam konteks otomatisasi industri, pengukuran dimensi produk, *quality control*, dan sortir otomatis memerlukan segmentasi yang akurat untuk pengambilan keputusan *real-time*. Sistem berbasis *bounding box* konvensional hanya memberikan perkiraan lokasi objek tanpa detail bentuk aktual, sehingga tidak memadai untuk aplikasi yang membutuhkan pengukuran presisi. *Instance segmentation* melalui Mask R-CNN menawarkan solusi dengan memberikan *mask* piksel-level yang dapat diintegrasikan dengan analisis spasial untuk estimasi ukuran dan volume objek secara otomatis. Eksplorasi mendalam terhadap performa Mask R-CNN menggunakan *framework* Detectron2 menjadi

krusial sebagai fondasi pengembangan sistem otomatisasi yang lebih kompleks dan akurat.

Implementasi *instance segmentation* menjadi fokus utama dalam penelitian ini karena kemampuannya untuk mengidentifikasi dan memisahkan tiap objek dalam sebuah gambar hingga ke tingkat pikselnya. Detectron2 merupakan salah satu *framework* yang dikembangkan oleh Facebook AI Research (FAIR) untuk tujuan deteksi dan segmentasi objek dengan teknik *deep learning*. Arsitektur Detectron2 bersifat modular dan fleksibel menjadi keunggulan *framework* ini sehingga memudahkan untuk penyesuaian terhadap struktur model (Richard et al., 2023), konfigurasi, maupun pipeline data. Detectron2 mendukung beberapa algoritma untuk *instance segmentation* salah satunya adalah algoritma Mask R-CNN. Algoritma ini memiliki keunggulan mampu memberikan hasil segmentasi objek secara piksel presisi (Shafa & Andono, 2025). Penerapan algoritma ini akan memberikan segmentasi objek yang jelas antara objek satu dengan lainnya dengan penggambaran bentuk (*mask*) sebagai *output* pada masing-masing objek yang dideteksi. Implementasi Detectron2 yang mendukung Mask R-CNN menjadi langkah penting pengembangan penelitian dalam bidang otomatisasi seperti prediksi ukuran pada suatu objek.

Perkembangan algoritma untuk melakukan segmentasi sangat banyak pada zaman sekarang (Pakpahan, 2021), namun secara spesifik untuk segmentasi yang bertujuan dalam otomatisasi pengukuran masih dapat dikatakan minim. Pemanfaatan algoritma Mask R-CNN yang bekerja secara *pixel to pixel* masih dapat digali lebih jauh terutama untuk pengukuran objek. Beberapa algoritma untuk mendeteksi objek memberikan *output* berupa *bounding box*, namun *output* ini hanya memberikan lokasi pada gambar letak suatu objek yang dideteksi (Maleh et al., 2023). Segmentasi objek yang akurat dapat berpotensi memberikan informasi yang lebih realistis dan informatif, sehingga dengan argumen ini dapat digunakan sebagai dasar atau landasan untuk pengembangan penelitian prediksi ukuran suatu objek.

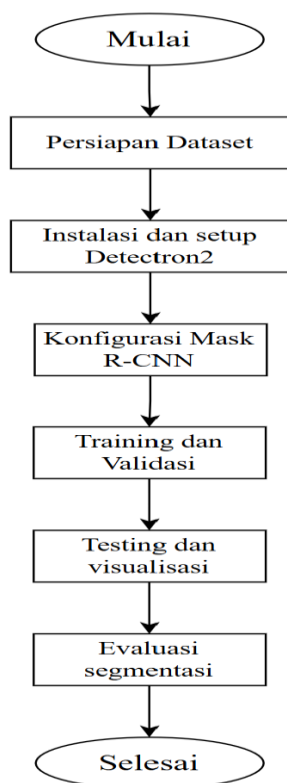
Perkembangan metode *instance segmentation* dalam beberapa tahun terakhir menunjukkan kemajuan yang sangat pesat dengan munculnya beragam arsitektur baru. Evolusi model dimulai dari Mask R-CNN dan terus berkembang ke varian-varian seperti Cascade Mask R-CNN, *Hybrid Task Cascade* (HTC), hingga model berbasis *transformer* seperti Mask2Former. Meskipun banyak model baru bermunculan, Mask R-CNN tetap menjadi *baseline* yang kuat karena mampu memberikan keseimbangan antara akurasi dan kebutuhan komputasi. Berbagai penelitian sebelumnya telah memanfaatkan Mask R-CNN untuk sejumlah aplikasi, misalnya deteksi cacat pengelasan (Ummah et al., n.d.), pengenalan makanan dan estimasi kalori (Shafa & Andono, 2025), serta analisis

objek medis. Framework Detectron2 dari Facebook AI Research (FAIR) menyediakan implementasi Mask R-CNN yang telah dioptimalkan dengan dukungan akselerasi GPU dan desain modular untuk memudahkan eksperimen. Meskipun demikian, masih terdapat kesenjangan penelitian yang perlu digali lebih jauh, antara lain eksplorasi konfigurasi *hyperparameter* yang paling sesuai untuk dataset kustom berukuran kecil hingga menengah, analisis lebih mendalam terhadap kasus kegagalan seperti objek tumpang tindih dan *under-detection*, serta strategi peningkatan performa untuk objek dengan variasi ukuran (kecil, sedang, besar). Penelitian ini berupaya menjawab kesenjangan tersebut dengan melakukan evaluasi menyeluruh terhadap performa Mask R-CNN serta mengidentifikasi *bottleneck* yang memengaruhi akurasi dalam sistem prediksi ukuran objek.

Penelitian ini bertujuan untuk implementasi dan evaluasi algoritma Mask R-CNN dengan menggunakan *framework* Detectron2 untuk deteksi dan segmentasi objek pada sebuah gambar. Fokus utama dalam penelitian ini untuk menguji kemampuan model yang dibangun dalam menghasilkan *instance segmentation* yang konsisten. Selain itu, penelitian ini diharapkan mampu memberikan pemahaman lebih dalam tentang performa yang dihasilkan oleh Mask R-CNN sebagai langkah awal dalam implementasinya untuk sistem yang lebih kompleks. Berdasarkan hasil segmentasi *pixel to pixel* oleh Mask R-CNN, penelitian ini diharapkan dapat memberikan peluang dalam integrasi analisis spasial untuk menghasilkan estimasi atau prediksi ukuran suatu objek yang lebih presisi dan akurat. Oleh karena itu, penelitian ini dilakukan sebagai langkah awal yang diharapkan dapat menjadi landasan untuk pengembangan penelitian lanjutan dalam prediksi ukuran objek maupun otomatisasi lainnya.

2. Metode

Penelitian yang dilakukan menggunakan pendekatan secara eksperimen untuk mengimplementasikan dan melakukan evaluasi terhadap performa Mask R-CNN yang diterapkan dalam *framework* Detectron2. Penggunaan metode ini akan digunakan sebagai landasan untuk perkembangan penelitian berikutnya sehingga dalam penelitian ini berfokus pada kemampuan metode dalam melakukan segmentasi objek. Adapun langkah penelitian yang dilakukan seperti pada Gambar 1 berikut.



Gambar 1. Langkah Eksplorasi Detectron2 dan Mask R-CNN

Pada Gambar 1 menunjukkan langkah-langkah yang dilakukan dalam penelitian untuk eksplorasi ini yang dimulai dari persiapan dataset. Dataset yang digunakan merupakan *custom dataset* dengan jumlah kurang lebih 200 gambar berupa *handphone* dan *flashdisk* sehingga saat implementasi akan menggunakan kedua kelas tersebut. Seluruh gambar pada dataset perlu dilakukan anotasi terlebih dahulu menggunakan format COCO (*Common Object in Context*) yang mendukung informasi seperti *bounding box* dan *mask polygon* (Supriadi et al., 2021). Adapun contoh dataset yang digunakan serta anotasinya ditunjukkan pada Gambar 2.



Gambar 2. Dataset dan Anotasinya

Dalam proses anotasi *tools* yang digunakan yaitu roboflow dengan pembagian rasio dataset menjadi 70% untuk proses *training*, 25% untuk proses *validation*, dan 5% untuk proses *testing*. Adapun spesifikasi dataset yang digunakan dalam penelitian ini ditunjukkan pada Tabel 1.

Tabel 1. Spesifikasi Dataset

Atribut	Deskripsi
Jumlah Total Citra	100
Kelas Objek	2 (<i>handphone, flashdisk</i>)
Format File	JPG/JPEG
Tipe Data	Image (RGB)
Resolusi	Bervariasi, 800-1200 piksel (normalisasi saat <i>preprocessing</i>)
Format Anotasi	COCO JSON (<i>bounding box + polygon mask</i>)
Jumlah <i>Training Set</i>	70 Citra (70%)
Jumlah <i>Validation Set</i>	25 Citra (25%)
Jumlah <i>Testing Set</i>	5 Citra (5%)

Pemilihan rasio pembagian dataset 70:25:5 didasarkan pada karakteristik dan keterbatasan jumlah data yang tersedia. Dari 100 citra yang dikumpulkan, sebanyak 70 citra dialokasikan untuk proses *training* dengan pertimbangan bahwa model *deep learning* memerlukan jumlah data latih yang memadai untuk dapat mempelajari pola dan fitur objek secara optimal. Meskipun jumlah ini tergolong kecil untuk standar *deep learning* pada umumnya, namun sudah cukup untuk memberikan variasi pose objek, kondisi pencahayaan, dan orientasi yang berbeda-beda dalam proses pembelajaran model.

Porsi *validation set* ditetapkan sebesar 25% atau setara 25 citra, yang memang lebih besar dibandingkan proporsi standar pada umumnya yang berkisar 10-20%. Keputusan ini diambil karena pada dataset berukuran kecil, *validation set* yang lebih besar sangat diperlukan untuk memantau performa model secara lebih akurat selama proses *training* berlangsung. Dengan *validation set* yang cukup besar, perhitungan metrik evaluasi seperti *loss* dan mAP menjadi lebih reliabel sehingga dapat digunakan sebagai acuan yang baik untuk melakukan *early stopping* dan penyesuaian *hyperparameter* guna mencegah terjadinya *overfitting*.

Sementara itu, *testing set* hanya dialokasikan 5% atau 5 citra yang memang terlihat sangat kecil. Namun demikian, jumlah ini masih dapat diterima untuk penelitian eksploratif tahap awal seperti ini karena fokus utama penelitian adalah untuk mengevaluasi kemampuan dasar model dalam melakukan segmentasi objek. *Testing set* ini tetap berfungsi sebagai data yang benar-benar belum pernah dilihat model (*unseen data*) untuk validasi akhir, dan dengan keterbatasan total 100 citra yang ada, lebih baik mengalokasikan porsi lebih besar untuk *training* dan *validation* agar model dapat belajar dengan lebih baik.

Berdasarkan literatur (Goodfellow et al., 2018), pembagian dengan proporsi *validation* yang lebih besar memang direkomendasikan untuk dataset kecil karena dapat memberikan stabilitas lebih baik dalam evaluasi model. Pada penelitian dengan jumlah data lebih besar, biasanya digunakan rasio seperti 80:10:10 atau 70:20:10 yang lebih standar. Keterbatasan jumlah data sebanyak 100 citra ini memang menjadi salah satu kelemahan dalam penelitian ini, sehingga ke depannya diperlukan pengumpulan data yang lebih banyak untuk

meningkatkan performa dan generalisasi model pada penelitian lanjutan.

Langkah berikutnya adalah konfigurasi *environment* pada Anaconda agar bisa mendukung *framework* yang digunakan yaitu Detectron2. *Framework* Detectron2 ini dibangun dengan PyTorch sebagai landasan algoritmanya dan dirancang dengan arsitektur modular sehingga mempermudah dalam penyesuaian eksperimen dan model yang dibangun (Abiamamela Obi-Obuoha et al., 2024). Konfigurasi *environment* dilakukan dengan instalasi *package* yang dibutuhkan untuk menunjang Detectron2 dengan Mask R-CNN (Facebook Research, 2021). Setelah konfigurasi *environment* selesai dilakukan, maka langkah selanjutnya adalah melakukan konfigurasi untuk Mask R-CNN. Algoritma ini merupakan salah satu model deteksi objek berdasarkan algoritma CNN (*Convolutional Neural Network*) dan merupakan perkembangan dari algoritma sebelumnya yaitu Faster R-CNN (Sanusi et al., 2023). Dalam proses konfigurasinya perlu menentukan beberapa parameter penting untuk menunjang proses training dari model yang dibangun. Adapun *hyperparameter* yang digunakan dalam konfigurasi Mask R-CNN pada penelitian ini ditunjukkan pada Tabel 2.

Tabel 2. Hyperparameter Mask R-CNN

Hyperparameter	Value
BASE_LR	0.02
MAX_ITER	270000
BATCH_SIZE_PER_IMAGE	256
BACKBONE	build_resnet_fpn_backbone
ROI_HEADS.SCORE_THRES	0.5
H TEST	

Pada penelitian ini Mask R-CNN dikonfigurasi dengan menggunakan beberapa parameter utama untuk mendapatkan hasil performa yang optimal berdasarkan *default* implementasi Detectron2 dengan Mask R-CNN. *Learning rate* ditetapkan sebesar 0.02, nilai ini terbilang cukup tinggi sehingga perlu memantau kestabilan *loss* selama proses *training*. Jumlah iterasi maksimum yang diterapkan sebesar 270.000 dengan jumlah *Region of Interest* (ROI) sebesar 256 yang diproses tiap citra selama proses *training*. Kedua parameter tersebut berpengaruh terhadap keseimbangan antara data positif dan negatif. *Backbone* Resnet_FPN digunakan dalam model yang dibangun karena memungkinkan untuk deteksi objek kecil, sedang, dan besar secara simultan serta penggunaan FPN dapat membantu mempertahankan informasi spasial dari resolusi yang bervariasi. Ambang batas nilai *confidence* ditentukan sebesar 0.5 yang berarti hanya prediksi dengan nilai diatas 50% saja yang ditampilkan untuk menjaga kestabilan antara presisi dan *recall* pada hasil segmentasi.

Tahapan berikutnya adalah melakukan *testing* dan visualisasi berdasarkan model yang telah dibangun. Proses *testing* dilakukan dengan menggunakan citra yang tidak ada pada dataset saat training dengan tujuan untuk menguji kemampuan

model. Hasil *output* dari model akan divisualisasikan dalam bentuk citra dengan *overlay mask* dan nama kelas pada setiap objek yang dideteksi. Proses evaluasi dilakukan dengan menghitung nilai *Mean Average Precision* (mAP) yang merupakan salah satu standar dalam menilai performa dari model (Palupi et al., 2023). Adapun rumus mAP yang digunakan sebagai berikut.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (1)$$

Metrik evaluasi ini merupakan nilai rata-rata dari *Average Precision* (AP) pada berbagai tingkat *recall* dan ambang batas *Intersection over Union* (IoU) (Cheng et al., n.d.). Pada perhitungan tersebut untuk model Mask R-CNN dilakukan dengan menggunakan protokol COCO, yaitu dengan menghitung rata-rata AP pada 10 ambang batas IoU dari 0.5 hingga 0.95 dengan interval sebesar 0.05.

3. Hasil dan Pembahasan

Pada bagian ini akan membahas hasil dari implementasi dan evaluasi model Mask R-CNN menggunakan *framework* Detectron2. Proses *training* model yang dibangun telah disesuaikan dengan konfigurasi parameter sebelumnya dan model diuji dengan data uji untuk mengukur performa segmentasi. Setelah itu, evaluasi dilakukan berdasarkan hasil uji visualisasi segmentasi yang diberikan oleh model serta *Mean Average Precision* (mAP) akan diterapkan untuk mengetahui kinerja model. Hasil ini akan menjadi dasar untuk perbaikan model dan pengembangan penelitian.

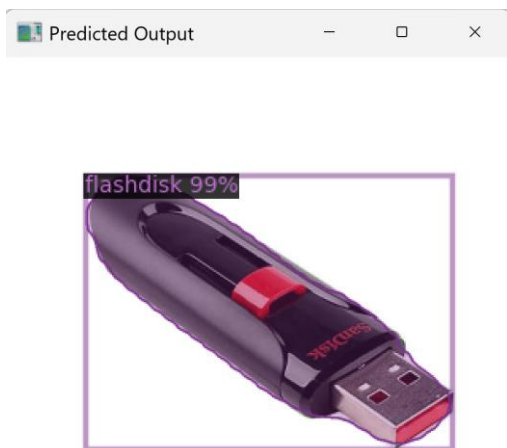
3.1. Visualisasi Hasil Segmentasi

Pengujian hasil segmentasi dilakukan berdasarkan model yang dibangun menggunakan dataset pribadi dan COCO dataset sebagai *pre-trained* model. Pemanfaatan bobot dari COCO dataset dan implementasi *transfer learning* dapat mempercepat proses *training*.



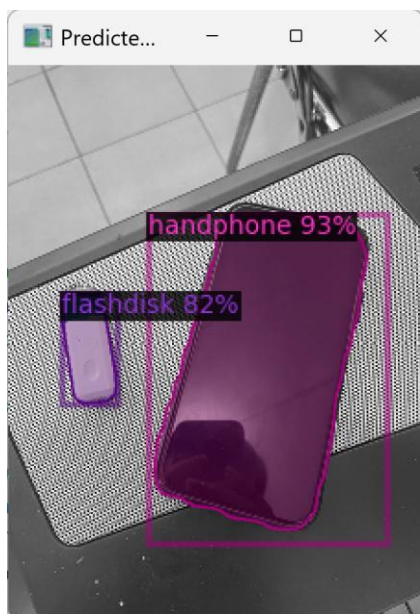
Gambar 3. Hasil Pengujian 1

Hasil pengujian pada Gambar 3 menunjukkan bahwa model berhasil mendeteksi objek. Model berhasil memberikan *bounding box* dan *mask* pada masing-masing objek dengan memberikan warna yang berbeda untuk setiap objeknya. Model juga menunjukkan presentase *confidence* sebesar 100% untuk setiap objek yang dideteksi.



Gambar 4. Hasil Pengujian 2

Pengujian pada Gambar 4 menunjukkan bahwa model berhasil mendeteksi objek. Model berhasil memberikan *bounding box* dan *mask* pada objek yang berupa *flashdisk* sesuai dengan nama kelas yang ditampilkan. Model juga menunjukkan presentase *confidence* sebesar 99% untuk objek yang dideteksi.



Gambar 5. Hasil Pengujian 3

Hasil pengujian pada Gambar 5 menunjukkan bahwa model berhasil mendeteksi objek yang sesuai dengan objek aktualnya yaitu *handphone* dan *flashdisk*. Model berhasil memberikan *bounding box*

dan *mask* pada masing-masing objek dengan memberikan warna yang berbeda untuk setiap objeknya serta penggambaran *mask* yang cukup sesuai dengan *edge* dari objek. Model juga menunjukkan presentase *confidence* sebesar 82% untuk objek dengan kelas *flashdisk* dan sebesar 93% untuk objek dengan kelas *handphone*.



Gambar 6. Hasil Pengujian 4

Pengujian selanjutnya ditunjukkan pada Gambar 6, namun terdapat kesalahan deteksi pada pengujian ini. Hasil deteksi menunjukkan dua *bounding box* dan *mask* dengan warna abu-abu dan *orange* yang seharusnya terdapat lebih dari dua objek yang dideteksi. *Mask* dengan warna *orange* menunjukkan nilai *confidence* sebesar 99%, namun seharusnya ada 4 objek sedangkan model hanya mendeteksi 1 objek saja. Hal ini dapat terjadi karena beberapa hal seperti terjadinya *overlapping* karena model tidak berhasil membedakan objek satu dengan lainnya. Lalu model juga mengalami *under-detection* dimana kondisi ini terjadi karena model menampilkan hasil deteksi dari *false negative*. Selain itu, dapat dikatakan bahwa model masih lemah untuk mendeteksi objek yang saling tumpang tindih atau identik.



Gambar 7. Hasil Pengujian 5

Pada Gambar 7 menunjukkan hasil percobaan yang mengalami *under-detection* karena pada gambar

terdapat 4 objek namun hanya 3 objek yang terdeteksi. Hasil deteksi seperti ini dapat terjadi oleh beberapa hal seperti pada Detectron2 menerapkan algoritma *Non-Maximum Supression* (NMS) yang berfungsi untuk menghilangkan deteksi yang berlebihan (*redundant*), sehingga objek yang tidak terdeteksi dapat dikatakan memiliki nilai yang sama dengan hasil deteksi didekatnya. Oleh karena itu, model membuang atau menghilangkan *bounding box* maka hasil visualisasi juga tidak menunjukkan hasil deteksi yang diharapkan.

3.2. Evaluasi Model

Evaluasi model dilakukan untuk mengetahui performa yang dihasilkan oleh model dalam mendeteksi maupun segmentasi pada gambar yang diuji. Pengukuran evaluasi dilakukan dengan menggunakan metrik *Mean Average Precision* (mAP) yang merupakan standar evaluasi model. Selain itu, nilai AP50 dan AP75 juga disajikan untuk memberikan gambaran performa model pada Tingkat toleransi yang lebih spesifik. Hasil evaluasi ini menjadi acuan dalam menilai performa model serta menentukan aspek yang dapat ditingkatkan untuk pengembangan penelitian selanjutnya.

Tabel 3. Evaluasi Model Average Precision

AP	AP50	AP75	API
62.615 %	71.836 %	67.884 %	62.615 %

Hasil evaluasi model pada Tabel 3 menunjukkan beberapa parameter dengan tingkat persentasenya tersendiri. Parameter AP (*Average Precision*) menunjukkan rata-rata presisi pada berbagai *threshold* IoU, selanjutnya terdapat parameter AP50 yang menunjukkan tingkat presisi deteksi saat *threshold* IoU bernilai 0.5 dan parameter AP75 menunjukkan tingkat presisi deteksi saat *threshold* IoU lebih ketat dalam proses deteksi dengan nilai 0.75. Parameter terakhir yang ditunjukkan adalah *Average Precision Large* (API) yang memberikan presentase bahwa model dapat mendeteksi objek yang besar, hal ini menjelaskan karena tidak ada parameter *Average Precision Small* (APs) dan *Average Precision Medium* (APm) maka model hanya bisa mendeteksi objek yang berukuran besar saja.

Berdasarkan AP yang bernilai 62.6% menunjukkan bahwa model yang dibangun cukup baik dalam mendeteksi mayoritas objek secara akurat walaupun model ini belum dilakukan *finetune* secara menyeluruh. Pada AP50 memiliki nilai 71.8% menunjukkan model cukup baik dalam deteksi objek dengan memberikan *bounding box* dan *mask* pada objek dengan nilai *threshold* sebesar 0.5 selama proses pengujian. Selanjutnya, pada AP75 memiliki nilai 67.8% menunjukkan bahwa model cukup bagus dalam mendeteksi objek dengan nilai *threshold* yang lebih tinggi yaitu 0.75 dalam proses pengujiannya.

AP50 mengukur presisi deteksi pada *threshold* IoU 0,5 yang menandakan model dianggap benar jika *overlap* antara prediksi dan *ground truth* diatas 50%. Nilai tertinggi pada AP50 menunjukkan bahwa model mampu mendeteksi Lokasi umum objek dengan akurat, meskipun *mask* yang dihasilkan belum sempurna menempel pada *boundary* objek. IoU sebesar 0.5 merupakan kriteria yang relative longgar, sehingga prediksi dengan *mask* yang kurang presisi masih dihitung sebagai *true positive*. Hal ini menjelaskan penyebab nilai AP50 lebih tinggi dibandingkan dengan AP75. Nilai AP50 yang tinggi menindikasikan bahwa model lebih baik dalam aspek deteksi (menemukan objek) dibandingkan segmentasi dengan tingkat presisi yang tinggi (menggambar *mask* yang *perfect-fit*).

AP (*Average Precision*) merupakan rata-rata dari AP pada berbagai *threshold* IoU yaitu 0.5 hingga 0.95 dengan interval 0.05, sedangkan Apl khusus diterapkan pada objek besar. Kedua parameter ini memiliki nilai terendah disebabkan oleh kondisi saat *threshold* seperti 0.75, 0.85, dan 0.95, model mengalami penurunan performa signifikan karena *mask* yang dihasilkan tidak cukup presisi pada *boundary* objek. Objek berukuran besar dalam dataset seperti *handphone* memiliki *edge* yang panjang. Ketidakpresisian kecil pada *edge detection* mengakibatkan area *error* yang lebih besar secara proporsional, sehingga menurunkan IoU.

Implikasi untuk pengembangan berdasarkan *gap* antara AP50 sebesar 71.8% dan AP75 sebesar 67.8% yang relatif kecil yaitu sebesar 4% menunjukkan adanya potensi dari model yang dibangun, namun nilai AP menurun dengan rata-rata sebesar 62.6%. Hal ini mengindikasikan perlunya *refinement* pada *mask prediction head* untuk presisi *boundary* yang lebih baik, augmentasi data dengan variasi skala objek (*small, medium, large*), dan *fine tuning* parameter NMS dan *anchor boxes* untuk deteksi yang lebih akurat pada *threshold* tinggi.

4. Kesimpulan

Berdasarkan hasil pelatihan dan evaluasi algoritma Mask R-CNN menggunakan *framework* Detectron2 pada dataset *custom* berisi 100 citra (2 kelas objek), dapat disimpulkan bahwa model yang dibangun cukup baik dalam melakukan *instance segmentation* dengan *output* berupa *bounding box* dan *mask* pada setiap objek yang dideteksi. Hal ini dinilai berdasarkan metrik evaluasi mAP yaitu AP 62.615%, AP50 71.836%, AP75 67.884%, dan API 62.615%.

Hasil evaluasi tersebut menandakan bahwa model memiliki kemampuan yang baik dalam mendeteksi lokasi objek dengan nilai AP50 mencapai 71.836%, namun ketika *threshold* IoU dinaikkan menjadi lebih ketat, nilai AP rata-rata turun menjadi 62.615% yang menandakan masih ada ruang perbaikan dalam hal presisi segmentasi. Selain itu,

model hanya mampu mendeteksi objek berukuran besar saja dengan nilai API 62.615%, sedangkan untuk objek kecil dan sedang tidak tersedia nilainya karena dataset yang digunakan kurang bervariasi dalam hal skala objek. Dari pengamatan secara visual terhadap hasil segmentasi, ditemukan beberapa permasalahan yang muncul selama proses pengujian. Masalah pertama adalah model mengalami kesulitan ketika harus mendeteksi objek-objek yang letaknya berdekatan atau tumpang tindih, hal ini kemungkinan terjadi karena mekanisme *Non-Maximum Suppression* yang diterapkan terlalu ketat dalam memfilter hasil deteksi. Masalah kedua adalah terjadinya *under-detection* dimana beberapa objek tidak terdeteksi sama sekali, terutama objek-objek yang memiliki nilai *confidence* yang rendah sehingga tidak ditampilkan dalam hasil akhir.

Berdasarkan temuan-temuan tersebut, terdapat beberapa hal yang perlu dilakukan untuk meningkatkan performa model pada penelitian berikutnya. Penambahan jumlah dataset dari 100 citra menjadi 500-1000 citra agar model memiliki data pembelajaran yang lebih banyak dan bervariasi sehingga dapat mengurangi kemungkinan *overfitting*. Selain itu, perlu dilakukan penyesuaian pada parameter NMS *threshold* dan ukuran *anchor box* agar lebih cocok dengan karakteristik objek yang akan dideteksi. Penerapan teknik augmentasi data seperti *flipping*, rotasi, *scaling*, dan *occlusion* juga dapat membantu memperkaya variasi data. Dataset juga perlu ditambahkan dengan citra yang memuat objek berukuran kecil hingga sedang agar model dapat dilatih untuk mendeteksi objek dengan berbagai ukuran. Beberapa hal lain yang bisa dieksplorasi adalah penggunaan arsitektur *backbone* yang berbeda seperti ResNet-101 atau EfficientNet untuk ekstraksi fitur yang lebih baik, serta penerapan teknik *post-processing* untuk menangani kasus objek yang saling tumpang tindih. Upaya perbaikan ini sangat penting untuk dilakukan guna mendukung pengembangan sistem prediksi ukuran objek yang lebih akurat pada tahap penelitian selanjutnya.

Daftar Pustaka:

- Abiamamela Obi-Obuoha, Victor Samuel Rizama, Ifeanyichukwu Okafor, Haggai Edore Ovwenkeper, Kehinde Obe, & Jeremiah Ekundayo. (2024). Real-time traffic object detection using detectron 2 with faster R-CNN. *World Journal of Advanced Research and Reviews*, 24(2), 2173–2189. <https://doi.org/10.30574/wjarr.2024.24.2.3559>
- Cheng, B., Girshick, R., Dollár, P., Berg, A. C., & Kirillov, A. (n.d.). *Boundary IoU: Improving Object-Centric Image Segmentation Evaluation*. <https://bowenc0221.github.io/boundary-iou>
- Facebook Research. (2021). *detectron2*. github. <https://github.com/facebookresearch/detectron2>
- Goodfellow, I., Bengio, Y., & Courville, A. (2018). Deep learning. *Genetic Programming and Evolvable Machines*, 19(1–2), 305–307. <https://doi.org/10.1007/s10710-017-9314-z>
- Hermansyah, M., & Fadillah, N. (2019). *Virtual Mouse Berdasarkan Warna RGB Menggunakan Metode Optical Flow Secara Real-Time*. 21(2), 249–252. <https://doi.org/10.31294/p.v20i2>
- Mahdiyah, L., Oktamuliani, S., & Putri, W. L. (2025). Penerapan Algoritma Deep Learning YOLOv8 pada Platform Roboflow untuk Segmentasi Citra Panoramik. *Jurnal Fisika Unand*, 14(3), 228–234. <https://doi.org/10.25077/jfu.14.3.228-234.2025>
- Maleh, I. M. D., Teguh, R., Sahay, A. S., Okta, S., & Pratama, M. P. (2023). Implementasi Algoritma You Only Look Once (YOLO) Untuk Object Detection Sarang Orang Utan Di Taman Nasional Sebangau. *Jurnal Informatika*, 10(1), 19–27. <https://doi.org/10.31294/inf.v10i1.13922>
- Pakpahan, R. (2021). ANALISA PENGARUH IMPLEMENTASI ARTIFICIAL INTELLIGENCE DALAM KEHIDUPAN MANUSIA. *Journal of Information System, Informatics and Computing Issue Period*, 5(2), 506–513. <https://doi.org/10.52362/jisicom.v5i2.616>
- Palupi, L., Ihsanto, E., & Nugroho, F. (2023). Analisis Validasi dan Evaluasi Model Deteksi Objek Varian Jahe Menggunakan Algoritma YOLOv5. *Journal of Information System Research (JOSH)*, 5(1), 234–241. <https://doi.org/10.47065/josh.v5i1.4380>
- Sanusi, A. P., Fariza, A., & Setiawardhana. (2023). Klasifikasi Tinggi Badan Menggunakan Metode Mask R-CNN. *Indonesian Journal of Computer Science Attribution*, 12(4), 2050.
- Rasyad, A. M. (2025). *Segmentasi Semantik Menggunakan Model DeepLabv3+ untuk Citra Satelit pada Area Tutupan Lahan dan Objek Gunung Api*.
- Richard, H., Natadjaja, A., & Wonohadidjojo, D. M. (2023). Penerapan Algoritma YOLOv4-Tiny Dan Efficientnetv2-S Untuk Deteksi Kesegaran Ikan Gurami. *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, 8(2), 480–488. <https://tunasbangsa.ac.id/ejurnal/index.php/juristik>
- Shafa, R. A., & Andono, P. N. (2025). PENDETEKSI VISUAL MAKANAN DAN JUMLAH KALORINYA MENGGUNAKAN ALGORITMA MASK R-CNN BERBASIS BOT TELEGRAM. *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 10(1), 641–651. <https://doi.org/10.29100/jupi.v10i1.6972>

- Supiyandi Supiyandi, & Rafif Rasendriya. (2024). Penggunaan Visi Komputer untuk Mengidentifikasi Jenis Buah dari Gambar. *Router : Jurnal Teknik Informatika Dan Terapan*, 2(4), 94–103. <https://doi.org/10.62951/router.v2i4.287>
- Supriadi, M. F., Rachmawati, E., Arifianto, A., & Korespondensi, P. (2021). *PEMBANGUNAN APLIKASI MOBILE PENGENALAN OBJEK UNTUK PENDIDIKAN ANAK USIA DINI*. 8(2), 357–364. <https://doi.org/10.25126/jtiik.202184363>
- Ummah, M., Sasmito, A., Tri Wahyuningrum, R., Imron Mustajib, M., Rizal, N., Eko Sandjaja, I., Triputrasyo Murwatono, T., Machfudin, A., Purnama Sari, D., Riset Teknologi Rancang Bangun Perkapalan dan Bangunan Apung Badan Riset dan Inovasi Nasional Surabaya, K., & Teknik, F. (n.d.). *Implementasi Mask R-CNN untuk Identifikasi Cacat pada Pengelasan Shielded Metal Arc Welding (SMAW)*.
- Vidia, S., Afni, N., Putri Silmina, E., Pangestu, I. B., Program, }, Teknologi, S., Sains, I. F., Universitas ', T., & Yogyakarta, A. (n.d.). Computer Vision digunakan untuk Memantau Pemuda di Masa Pandemi Covid-19. In *Procedia of Engineering and Life Science* (Vol. 1, Issue 2).
- Yunita, H., & Setyati, E. (2019). Hand Gesture Recognition Sebagai Pengganti Mouse Komputer Menggunakan Kamera. *Jurnal ELTIKOM*, 3(2), 64–76. <https://doi.org/10.31961/eltikom.v3i2.114>