

PENGEMBANGAN SISTEM PENDETEKSI DAN PENGHITUNGAN UANG RUPIAH OTOMATIS MENGGUNAKAN YOLOV8

Achmad Fauzal Khobir¹, Muthiah Hamidah², Mayanda Mega Santoni³, Dodik Ariyanto⁴, Faldiena Marcelita⁵

Program Studi Teknologi Rekayasa Perangkat Lunak, Sekolah Vokasi, IPB University^{1, 2}
Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jakarta³
Program Studi Teknologi Rekayasa Komputer, Sekolah Vokasi, IPB University^{4, 5}
Kampus IPB Cilibende, Jl. Kumbang No.14, Kota Bogor, Jawa Barat, Indonesia^{1, 2, 4, 5}
E-mail: achfauzal@apps.ipb.ac.id¹, muthiahhamidah@apps.ipb.ac.id², megasantoni@upnvj.ac.id³,
dodikariyanto@apps.ipb.ac.id⁴, faldiena.m@apps.ipb.ac.id⁵

Abstrak - Perkembangan teknologi *computer vision* dan *deep learning* telah mendorong pengembangan sistem deteksi objek secara otomatis pada berbagai bidang, termasuk pengenalan mata uang. Penelitian ini bertujuan untuk mengembangkan sistem pendeteksi dan penghitungan uang rupiah otomatis menggunakan algoritma YOLOv8. Penelitian dilakukan dengan membandingkan performa dua varian model, yaitu YOLOv8n dan YOLOv8s, dalam mendeteksi nominal uang rupiah berdasarkan metrik evaluasi *precision*, *recall*, mAP50, dan mAP50-95. Tahapan penelitian meliputi pengumpulan dataset, preprocessing data, pelatihan model, pengujian model, evaluasi kinerja, serta analisis hasil. Dataset yang digunakan terdiri dari citra uang rupiah dengan tujuh kelas nominal, yaitu Rp1.000, Rp2.000, Rp5.000, Rp10.000, Rp20.000, Rp50.000, dan Rp100.000. Hasil penelitian menunjukkan bahwa kedua model mampu mendeteksi nominal uang rupiah dengan baik serta melakukan penghitungan total nominal uang secara otomatis. Model YOLOv8s menunjukkan performa yang lebih baik dalam mendeteksi objek secara umum dan menghasilkan deteksi yang lebih akurat, sedangkan YOLOv8n memiliki kemampuan lokalisasi objek yang lebih stabil berdasarkan nilai mAP50-95 yang lebih tinggi serta ukuran model yang lebih ringan. Beberapa kesalahan deteksi masih ditemukan pada kondisi uang yang bertumpuk (*overlap*), pencahayaan yang kurang optimal, dan kemiripan warna antar-nominal uang. Secara keseluruhan, penelitian ini menunjukkan bahwa algoritma YOLOv8 memiliki potensi yang baik untuk diterapkan pada sistem pendeteksi dan penghitungan uang rupiah otomatis berbasis *computer vision* serta dapat diimplementasikan pada aplikasi berbasis web.

Kata Kunci: *Computer Vision*, *Deep Learning*, Deteksi Objek, Uang Rupiah, YOLOv8

I. PENDAHULUAN

Perkembangan teknologi di bidang *computer vision* dan *deep learning* saat ini berkembang sangat pesat dan banyak dimanfaatkan dalam berbagai bidang, salah satunya adalah sistem deteksi objek secara otomatis. Secara fundamental, teknologi ini bertujuan untuk memberikan dampak positif berupa efektivitas dan efisiensi dalam mengolah data visual guna membantu berbagai aktivitas manusia sehari-hari (Azhar et al., 2021). Dalam sektor keuangan, pengolahan citra digital telah lama dikembangkan untuk mengenali karakteristik unik pada dokumen berharga, termasuk mata uang, guna memastikan kelancaran proses transaksi (Pambudi et al., 2020).

Uang kertas Rupiah merupakan instrumen utama dalam transaksi ekonomi masyarakat Indonesia yang memiliki keberagaman identitas visual pada setiap pecahannya (Rafikil et al., 2025). Untuk mengenali nominal tersebut secara otomatis, berbagai metode telah diuji mulai dari penggunaan fitur warna hingga deteksi tepi konvensional. Namun, pendekatan tradisional seperti penggunaan fitur warna RGB sering kali menghadapi kendala saat harus menangani uang dalam kondisi fisik yang

tidak ideal, seperti uang yang sudah lusuh, terlipat, atau saat pengambilan gambar dilakukan dalam kondisi pencahayaan yang minim (A. R. Pratama et al., 2020). Masalah tersebut diperparah oleh fakta bahwa fitur fisik seperti *blind code* sering kali sudah tidak dapat dirasakan lagi pada uang yang kondisinya sudah tidak baik (Rizkilillah et al., 2024).

Sebagai solusi modern, algoritma *You Only Look Once* (YOLO) kini menjadi standar baru dalam deteksi objek karena kemampuannya memproses gambar dengan kecepatan tinggi secara *real-time*. Versi terbaru, yaitu YOLOv8, membawa pembaruan pada arsitektur jaringan yang memungkinkan deteksi menjadi lebih akurat namun tetap ringan saat dijalankan pada perangkat dengan sumber daya terbatas (Iqbal et al., 2025). Keunggulan inilah yang membuat YOLOv8 sangat potensial untuk dikembangkan sebagai sistem pendeteksi mata uang yang praktis dan portabel.

Meskipun penelitian mengenai pengenalan mata uang menggunakan YOLOv8 sudah mulai bermunculan, sebagian besar studi masih memiliki keterbatasan yang cukup signifikan. Umumnya, peneliti hanya terpaku pada penggunaan satu varian model tanpa membandingkan performa antarvarian

yang tersedia (Iqbal et al., 2025). Selain itu, sistem yang dikembangkan saat ini biasanya hanya sebatas memberikan label nominal pada setiap lembar uang yang terdeteksi (Steven Immanuel Sihombing et al., 2024). Belum ditemukan adanya fitur yang mampu melakukan penghitungan secara keseluruhan untuk menjumlahkan total nilai dari seluruh uang yang tertangkap oleh kamera dalam satu waktu. Hal ini menjadi celah penting, karena dalam transaksi nyata, pengguna sering kali berhadapan dengan beberapa lembar uang sekaligus dan membutuhkan informasi jumlah total nilai uang secara cepat tanpa harus menghitungnya secara manual.

Berdasarkan kesenjangan tersebut, penelitian ini dilakukan untuk mengembangkan sistem pendeteksi dan penghitungan uang rupiah otomatis menggunakan YOLOv8. Penelitian ini memberikan kontribusi berupa studi komparasi antara arsitektur YOLOv8n (*nano*) dan YOLOv8s (*small*) untuk menganalisis varian model yang paling optimal dalam menyeimbangkan kecepatan proses dan ketepatan deteksi. Selain itu, penelitian ini juga mengimplementasikan fitur penghitungan otomatis yang mampu menjumlahkan total nominal uang yang terdeteksi secara *real-time*. Model hasil pelatihan kemudian diimplementasikan ke dalam *website* berbasis Laravel bernama RupiScan agar sistem dapat digunakan secara lebih interaktif, praktis, dan mudah diakses oleh pengguna.

II. TINJAUAN PUSTAKA

2.1 Computer Vision

Computer vision adalah salah satu cabang ilmu komputer yang berfokus pada pengembangan sistem yang dapat secara otomatis melihat, mengenali, dan memahami gambar dan video (Sutisna et al., 2024).

2.2 Deep Learning

Deep learning adalah salah satu cabang *machine learning* yang menggunakan *deep neural network* untuk menyelesaikan permasalahan pada *domain machine learning* (Wahyuni & Sulaeman, 2022).

2.3 Deteksi Objek

Metode deteksi objek adalah salah satu cabang dari *computer vision* yang digunakan untuk menemukan objek dalam gambar menggunakan *bounding box* dan mengklasifikasikan objek ke dalam kelas tertentu (M. D. R. Pratama et al., 2022).

2.4 Algoritma YOLO

Algoritma *You Only Look Once (YOLO)* adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara *real-time*. Sistem pendeteksian yang dilakukan adalah dengan menggunakan *repurpose classifier* atau *localizer*

untuk melakukan deteksi (Rachmawati & Widhyaestoeti, n.d.). YOLO menggunakan jaringan saraf tiruan untuk mendeteksi objek, serta dapat digunakan untuk menghitung jumlah objek pada video atau gambar (Akbar & Rahman, 2025).

2.4.1 YOLOv8n

YOLOv8 hadir dalam beberapa varian, salah satunya adalah YOLOv8n. YOLOv8n merupakan model paling ringan dengan jumlah parameter paling kecil, sehingga cocok untuk kebutuhan inferensi cepat di perangkat dengan daya komputasi terbatas (Tanujaya et al., 2026).

2.4.2 YOLOv8s

YOLOv8s merupakan varian YOLOv8 yang memberikan keseimbangan antara efisiensi dan akurasi, sehingga model ini mampu menghasilkan performa deteksi objek yang stabil dengan kebutuhan komputasi yang tetap optimal (Taufiqurrahman et al., 2024).

2.5 Penelitian Terdahulu

Beberapa penelitian sebelumnya telah dilakukan untuk membandingkan performa model YOLOv8n dan YOLOv8s. Penelitian berjudul “Analisis Komparatif Model YOLOv8n dan YOLOv8s untuk Deteksi Objek Berdasarkan Metrik Evaluasi dan Confidence Score” oleh Upuy et al. (2026) (Upuy et al., 2026) menunjukkan bahwa YOLOv8n memiliki keunggulan pada efisiensi komputasi dengan waktu pelatihan yang lebih cepat. Sementara itu, YOLOv8s menunjukkan performa yang lebih baik pada metrik *recall*, mAP0.5, serta menghasilkan nilai confidence dan CRI yang lebih tinggi. Hasil tersebut menunjukkan bahwa YOLOv8s memberikan hasil deteksi yang lebih stabil dan konsisten, sedangkan YOLOv8n lebih unggul dalam aspek efisiensi komputasi.

Sejalan dengan penelitian tersebut, Marchiningrum et al. (2026) (Marchiningrum et al., 2026) dalam penelitian berjudul “Deteksi Otomatis Kematangan Buah Kelapa Sawit Menggunakan YOLOv8 untuk Pemilihan Model Deteksi Terbaik” menyatakan bahwa YOLOv8s memiliki keunggulan dalam hal kestabilan hasil dan ketepatan identifikasi pada berbagai kategori objek, sehingga dinilai lebih adaptif untuk diterapkan pada kondisi operasional dengan keterbatasan sumber daya komputasi. Sementara itu, YOLOv8n menawarkan kecepatan proses yang lebih tinggi sehingga lebih sesuai digunakan pada aplikasi *real-time* dengan sumber daya terbatas.

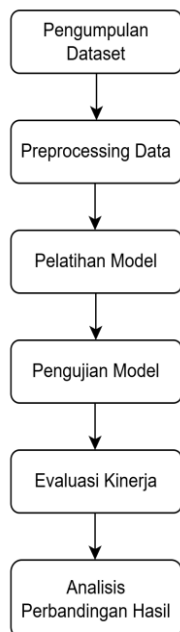
III. METODE PENELITIAN

3.1 Metode Penelitian

Penelitian ini menerapkan metode eksperimen untuk menganalisis dan membandingkan performa

dua model deteksi objek, yaitu YOLOv8n dan YOLOv8s. Pendekatan eksperimen dilakukan dengan melatih kedua model menggunakan dataset yang sama, kemudian mengevaluasi kinerjanya berdasarkan beberapa metrik evaluasi, seperti tingkat akurasi dan ketepatan deteksi.

3.2 Alur Penelitian



Gambar 1. Alur Penelitian

Alur penelitian yang digunakan pada penelitian ini ditunjukkan pada Gambar 1. Alur tersebut menggambarkan tahapan penelitian yang dilakukan secara sistematis dan berurutan mulai dari pengumpulan dataset, *preprocessing* data, pelatihan model, pengujian model, evaluasi kinerja, hingga analisis hasil.

3.2.1 Pengumpulan Dataset

Pengumpulan dataset merupakan langkah awal sebelum YOLO diajarkan untuk mengenal benda-benda yang harus dideteksi (Rachmawati & Widhyaestoeti, n.d.). *Dataset* yang digunakan dalam penelitian ini berasal dari *Roboflow Universe Money Detection Dataset* (*Money Detection Dataset*, n.d.) yang dapat dilihat pada Gambar 2. *Dataset* tersebut berisi citra uang kertas rupiah dengan tujuh kelas nominal, yaitu Rp1.000, Rp2.000, Rp5.000, Rp10.000, Rp20.000, Rp50.000, dan Rp100.000. *Dataset* dibagi menjadi tiga bagian, yaitu data training sebanyak 430 gambar, data *validation* sebanyak kurang lebih 40 gambar, dan data *test* sebanyak kurang lebih 20 gambar.



Gambar 2. *Dataset* Uang Rupiah

3.2.2 Preprocessing Data

Preprocessing data adalah teknik yang digunakan untuk mengubah data mentah menjadi format yang berguna dan efisien. Beberapa proses dalam data *preprocessing* adalah integrasi data, transformasi data, pembersihan data, dan reduksi data (Hamdani et al., 2024).

Pada penelitian ini, proses *preprocessing* diawali dengan mengonversi *file* anotasi dari format CSV ke format YOLO (.txt). Selanjutnya, dilakukan normalisasi *bounding box* agar koordinat objek sesuai dengan format yang dibutuhkan oleh model YOLOv8. Setelah proses normalisasi, *dataset* disusun ke dalam struktur *folder* yang terdiri dari *train/images* dan *train/labels* untuk data pelatihan, *valid/images* dan *valid/labels* untuk data validasi, serta *test/images* dan *test/labels* untuk data pengujian.

Tahap berikutnya adalah pembuatan file data.yaml yang berisi informasi kelas nominal uang yang digunakan pada penelitian ini, yaitu:

- 0 : Rp1.000
- 1 : Rp2.000
- 2 : Rp5.000
- 3 : Rp10.000
- 4 : Rp20.000
- 5 : Rp50.000
- 6 : Rp100.000

Selain itu, distribusi jumlah lembar uang pada dataset juga dicatat untuk mengetahui persebaran data pada setiap kelas nominal. Dataset dibagi menjadi data pelatihan (*train*), data validasi (*valid*), dan data pengujian (*test*). Distribusi jumlah data pada setiap kelas nominal uang ditunjukkan pada Tabel 1.

Tabel 1. Distribusi Dataset Berdasarkan Nominal Uang

Nominal	Train	Valid	Test
Rp 1.000	261	27	18
Rp 2.000	312	23	11
Rp 5.000	237	27	7
Rp 10.000	189	21	19
Rp 20.000	189	10	5
Rp 50.000	231	23	8
Rp100.000	174	17	8
Total	1.593	148	76

Perhitungan distribusi nominal uang dilakukan karena dalam satu gambar dapat terdapat beberapa lembar uang dengan nominal yang berbeda. Oleh karena itu, distribusi data dihitung berdasarkan jumlah objek pada setiap kelas nominal, bukan berdasarkan jumlah gambar. Distribusi ini digunakan untuk mengetahui persebaran data pada setiap kelas dan sebagai acuan dalam pembuatan *confusion matrix* untuk mengevaluasi performa model pada masing-masing nominal uang.

3.2.3 Pelatihan Model

Pelatihan model dilakukan menggunakan algoritma YOLOv8n dan YOLOv8s dengan dataset hasil *preprocessing*. Pada proses *training*, model dilatih selama 50 *epoch* untuk masing-masing algoritma dengan *batch size* 16 dan ukuran gambar sebesar 640×640 piksel. Proses pelatihan bertujuan agar model mampu mempelajari pola, bentuk, dan karakteristik dari setiap nominal uang rupiah yang terdapat pada *dataset*.

3.2.4 Pengujian Model

Setelah proses pelatihan selesai, model diuji menggunakan data testing dan validasi. Pengujian dilakukan untuk mengetahui kemampuan model dalam mendeteksi dan mengenali nominal uang rupiah secara otomatis pada kondisi yang berbeda dari data pelatihan. Pada tahap ini, model akan menghasilkan *bounding box*, label nominal uang, serta jumlah objek yang berhasil dideteksi pada gambar input. Selain mendeteksi nominal uang, sistem juga melakukan penghitungan total nilai uang berdasarkan hasil deteksi objek yang diperoleh.

3.2.5 Evaluasi Kinerja

Evaluasi kinerja model dilakukan untuk mengukur tingkat keberhasilan model dalam mendeteksi objek uang rupiah. Metrik evaluasi yang digunakan meliputi *precision*, *recall*, mAP50, dan mAP50-95. Nilai *precision* adalah nilai sensitifitas atau nilai ketepatan sistem antara informasi yang diberikan oleh sistem untuk menunjukkan secara benar data kelas negatif atau kelas positif (Azhari et al., 2021). Sedangkan *recall* adalah proporsi dari jumlah catatan yang dikembalikan oleh proses pencarian dalam sistem pengambilan data (Hapsari & Indrati, 2022). Selain itu, nilai mAP50 digunakan untuk mengukur akurasi deteksi objek pada *threshold Intersection over Union* (IoU) sebesar 0.5, sedangkan mAP50-95 digunakan untuk mengevaluasi performa model pada rentang *threshold IoU* 0.5 hingga 0.95. Semakin tinggi nilai mAP yang diperoleh, maka semakin baik kemampuan model dalam mendeteksi dan melokalisasi objek.

3.2.6 Analisis Perbandingan Hasil

Analisis hasil dilakukan berdasarkan nilai evaluasi model dan hasil deteksi pada data pengujian. Pada tahap ini dilakukan pengamatan terhadap kemampuan model dalam mendeteksi nominal uang, tingkat akurasi deteksi, serta kemampuan sistem dalam menghitung total nilai uang secara otomatis.

IV. HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian Model

Berdasarkan hasil pengujian yang telah dilakukan, model YOLOv8n dan YOLOv8s mampu mendeteksi nominal uang rupiah dengan baik. Pengujian dilakukan menggunakan dataset uji serta beberapa gambar baru di luar dataset *training* untuk mengetahui kemampuan model dalam mengenali objek uang pada berbagai kondisi.

Hasil pengujian menunjukkan bahwa kedua model mampu mendeteksi lebih dari satu lembar uang dalam satu gambar serta melakukan klasifikasi nominal uang secara otomatis. Selain itu, sistem juga berhasil menghitung total nominal uang berdasarkan hasil deteksi objek yang diperoleh.

Tabel 2. Perbandingan Kinerja YOLOv8n dan YOLOv8s

Model	<i>Precision</i>	<i>Recall</i>	mAP50	mAP50-95
YOLOv8n	0.91	0.87	0.94	0.80
YOLOv8s	0.91	0.88	0.95	0.71

Berdasarkan hasil pengujian pada Tabel 2, model YOLOv8n dan YOLOv8s menunjukkan performa yang cukup baik dalam mendeteksi nominal uang rupiah. Kedua model memiliki nilai *precision* yang sama, yaitu sebesar 0.91. Hal ini menunjukkan bahwa sebagian besar objek yang dideteksi oleh model merupakan objek yang benar dan tingkat kesalahan deteksi (*false positive*) relatif rendah.

Pada nilai *recall*, YOLOv8s memperoleh nilai sebesar 0.88, sedikit lebih tinggi dibandingkan dengan YOLOv8n sebesar 0.87. Hasil ini menunjukkan bahwa YOLOv8s memiliki kemampuan yang sedikit lebih baik dalam menemukan seluruh objek uang yang terdapat pada gambar. Dengan kata lain, kemungkinan objek tidak terdeteksi (*false negative*) pada YOLOv8s lebih kecil dibandingkan dengan YOLOv8n.

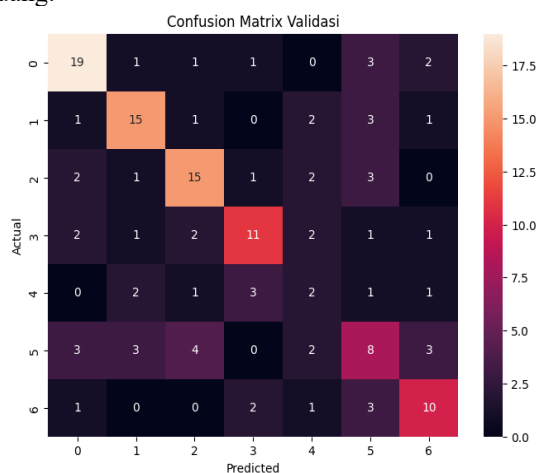
Dilihat dari nilai mAP50, model YOLOv8s memperoleh nilai sebesar 0.95, sedangkan YOLOv8n memperoleh nilai sebesar 0.94. Nilai tersebut menunjukkan bahwa kedua model memiliki tingkat akurasi deteksi objek yang tinggi pada *threshold IoU* 0.5. Namun, YOLOv8s memberikan

hasil yang sedikit lebih baik dalam mendeteksi dan mengklasifikasikan nominal uang rupiah.

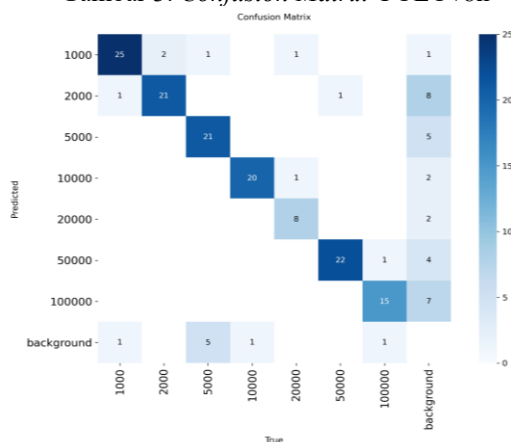
Sementara itu, pada metrik mAP50-95, YOLOv8n memperoleh nilai sebesar 0.80 dan YOLOv8s sebesar 0.71. Hasil ini menunjukkan bahwa YOLOv8n memiliki performa yang lebih stabil pada berbagai tingkat *threshold* IoU yang lebih ketat dibandingkan dengan YOLOv8s. Dengan demikian, YOLOv8n memiliki kemampuan lokalisasi objek yang lebih baik dalam menentukan *bounding box* secara lebih presisi. Secara keseluruhan, kedua model memiliki performa yang baik dengan keunggulan pada aspek yang berbeda.

4.2 Visualisasi Hasil

Confusion matrix data validasi ditunjukkan pada Gambar 3 dan Gambar 4. Berdasarkan hasil tersebut, model YOLOv8s menunjukkan tingkat deteksi yang lebih baik dibandingkan dengan model YOLOv8n, yang ditunjukkan oleh jumlah prediksi benar yang lebih tinggi pada setiap kelas nominal uang.



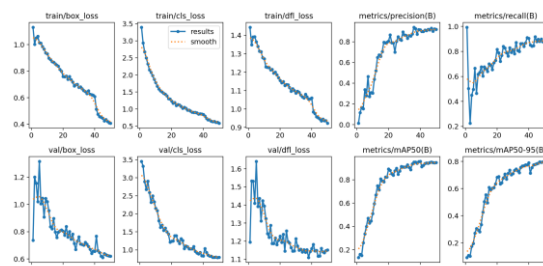
Gambar 3. *Confusion Matrix* YOLOv8n



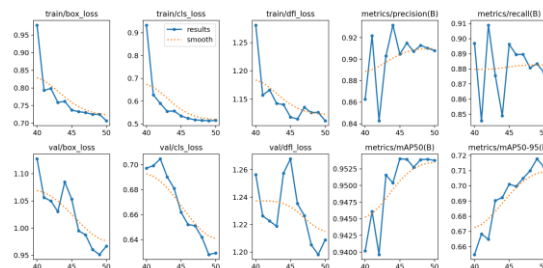
Gambar 4. *Confusion Matrix* YOLOv8s

Selain itu, grafik hasil pelatihan model YOLOv8n dan YOLOv8s juga ditampilkan pada Gambar 5 dan Gambar 6. Grafik tersebut menunjukkan perubahan nilai *loss* serta metrik

evaluasi selama proses pelatihan berlangsung. Berdasarkan grafik tersebut, terlihat bahwa nilai *train loss* dan *validation loss* pada kedua model mengalami penurunan seiring bertambahnya *epoch*. Hal ini menunjukkan bahwa model berhasil mempelajari pola objek uang pada dataset dengan cukup baik. Selain itu, nilai *precision*, *recall*, dan mAP pada kedua model juga mengalami peningkatan selama proses *training*, yang menandakan adanya peningkatan performa deteksi objek.



Gambar 5. Grafik Hasil Pelatihan YOLOv8n



Gambar 6. Grafik Hasil Pelatihan YOLOv8s

4.3 Percobaan Deteksi Uang

Percobaan deteksi uang dilakukan menggunakan model YOLOv8n dan YOLOv8s yang telah dilatih hingga *epoch* ke-50. Pengujian dilakukan menggunakan 10 citra uang Rupiah yang sama pada kedua model untuk membandingkan kemampuan deteksi dan klasifikasi nominal uang secara langsung. Hasil pengujian ditunjukkan pada Tabel 3, sedangkan contoh hasil deteksi dapat dilihat pada Gambar 5 dan Gambar 6.

Tabel 2. Pengujian Implementasi

Parameter	YOLOv8n	YOLOv8s
Jumlah Gambar Uji	10	10
Deteksi Benar	4	6
Deteksi Salah	6	4
Akurasi Implementasi	40%	60%

=== HASIL DETEKSI ===

Rp20,000 → 2 lembar = Rp40,000
 Rp1,000 → 3 lembar = Rp3,000
 Rp10,000 → 1 lembar = Rp10,000
 Rp5,000 → 1 lembar = Rp5,000
 Rp100,000 → 2 lembar = Rp200,000
 Rp50,000 → 1 lembar = Rp50,000
 Rp2,000 → 2 lembar = Rp4,000

TOTAL UANG: Rp312,000

DEBUG jumlah deteksi: 12



Gambar 5. Hasil Deteksi Objek Menggunakan YOLOv8n

=== HASIL DETEKSI ===

Rp1,000 → 1 lembar = Rp1,000
 Rp10,000 → 1 lembar = Rp10,000
 Rp5,000 → 1 lembar = Rp5,000
 Rp20,000 → 1 lembar = Rp20,000
 Rp2,000 → 3 lembar = Rp6,000
 Rp50,000 → 1 lembar = Rp50,000
 Rp100,000 → 1 lembar = Rp100,000

TOTAL UANG: Rp192,000

DEBUG jumlah deteksi: 9



Gambar 6. Hasil Deteksi Objek Menggunakan YOLOv8s

Berdasarkan hasil pengujian, model YOLOv8s menunjukkan tingkat keberhasilan deteksi yang lebih tinggi dibandingkan YOLOv8n. YOLOv8s mampu mengenali nominal uang dan menghitung total nilai uang dengan lebih konsisten pada sebagian besar citra uji. Sementara itu, YOLOv8n masih mengalami beberapa kesalahan deteksi yang menyebabkan hasil perhitungan total uang tidak sesuai dengan nilai sebenarnya.

Meskipun YOLOv8n memperoleh nilai mAP50-95 yang lebih tinggi pada tahap evaluasi model, hasil pengujian implementasi menunjukkan bahwa YOLOv8s memberikan performa yang lebih baik pada penggunaan nyata. Hal ini terlihat dari

kemampuan YOLOv8s dalam mendeteksi beberapa lembar uang dalam satu gambar dengan tingkat kesalahan yang lebih rendah dibandingkan dengan YOLOv8n.

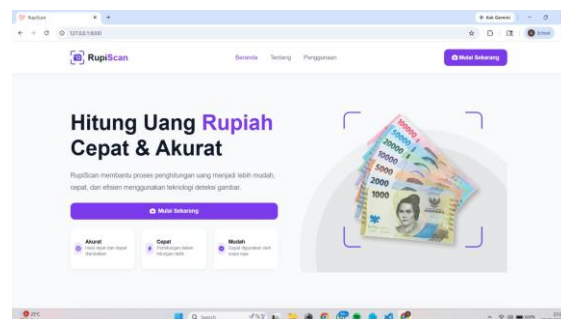
Beberapa kesalahan deteksi masih ditemukan pada kedua model. Kesalahan tersebut umumnya disebabkan oleh kondisi pencahayaan yang kurang optimal, kualitas citra yang rendah, serta adanya tumpang tindih antarlembar uang (occlusion). Pada kondisi tersebut, sebagian informasi visual uang tidak terlihat secara utuh sehingga model mengalami kesulitan dalam mengidentifikasi nominal dengan tepat.

4.4 Implementasi Sistem Berbasis Website

Untuk mempermudah penggunaan sistem deteksi uang rupiah secara langsung, penelitian ini juga mengimplementasikan model hasil pelatihan ke dalam website bernama RupisCan. Website dibangun menggunakan *framework* Laravel dan bahasa pemrograman PHP. Model yang digunakan pada implementasi sistem adalah YOLOv8s hasil pelatihan *epoch* ke-50 karena model tersebut memiliki performa deteksi yang paling baik dan akurat berdasarkan hasil penelitian yang telah dilakukan.

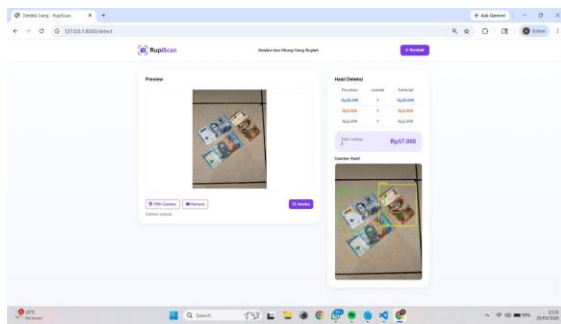
Website yang dikembangkan memiliki fitur utama berupa unggah gambar uang dan pengambilan foto uang secara langsung melalui kamera perangkat. Setelah gambar diproses oleh sistem, website akan menampilkan hasil deteksi berupa *bounding box* pada setiap uang yang terdeteksi, label nominal uang, serta total keseluruhan nilai uang secara otomatis.

Alur kerja sistem dimulai ketika pengguna mengunggah gambar atau mengambil foto uang melalui *website*. Selanjutnya, gambar akan dikirim ke model YOLOv8s untuk diproses. Hasil deteksi kemudian dikembalikan ke *website* dan ditampilkan dalam bentuk gambar hasil deteksi beserta informasi total nominal uang.



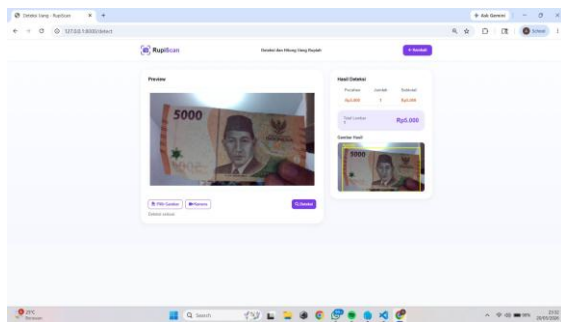
Gambar 7. Landing Page RupisCan

Gambar 7 merupakan tampilan *landing page website* RupisCan yang digunakan sebagai halaman utama sistem. Pada halaman ini, pengguna dapat mengakses fitur deteksi uang melalui menu yang tersedia.



Gambar 8. Percobaan Unggah Gambar

Gambar 8 menunjukkan proses ketika pengguna mengunggah gambar uang ke dalam sistem. Setelah gambar berhasil diproses, sistem akan menampilkan hasil deteksi nominal uang beserta total nilai uang yang berhasil dihitung secara otomatis.



Gambar 9. Percobaan Akses Kamera

Gambar 9 menunjukkan fitur deteksi menggunakan kamera secara langsung. Pengguna dapat mengarahkan kamera ke objek uang rupiah, kemudian sistem akan melakukan proses deteksi secara otomatis dan menampilkan hasil penghitungan total uang secara *real-time*.

Implementasi *website* ini bertujuan untuk menunjukkan bahwa sistem deteksi dan penghitungan uang rupiah otomatis tidak hanya dapat berjalan pada tahap eksperimen model, tetapi juga dapat diterapkan pada aplikasi nyata berbasis web.

4.5 Keterbatasan Penelitian

Penelitian ini memiliki beberapa keterbatasan yang perlu diperhatikan, yaitu jumlah dataset yang digunakan masih terbatas sehingga dapat memengaruhi kemampuan generalisasi model dalam menghadapi berbagai kondisi nyata. Variasi data seperti posisi uang, sudut pengambilan gambar, kondisi pencahayaan, serta tingkat kerusakan uang belum sepenuhnya terwakili dalam dataset yang digunakan.

Selain itu, pengujian model hanya dilakukan menggunakan dataset uji dan beberapa gambar baru, sehingga performa sistem belum diuji secara

menyeluruh pada data dalam jumlah besar maupun pada lingkungan nyata secara langsung. Selain itu, kondisi uang yang bertumpuk atau saling menutupi (*overlap*) masih menjadi kendala dalam proses deteksi. Pada beberapa kasus, model mengalami kesulitan dalam mengenali nominal uang ketika sebagian objek tertutup oleh lembar uang lainnya.

Penelitian ini juga hanya membandingkan dua varian model YOLOv8, yaitu YOLOv8n dan YOLOv8s, tanpa melibatkan metode deteksi objek lain sebagai pembanding. Oleh karena itu, belum dapat diketahui bagaimana performa model dibandingkan dengan algoritma deteksi objek lainnya.

4.6 Analisis Kesalahan

Berdasarkan hasil pengujian, ditemukan beberapa kesalahan deteksi pada model YOLOv8n dan YOLOv8s. Kesalahan tersebut umumnya terjadi pada kondisi pencahayaan rendah, objek uang yang saling bertumpuk (*overlapping*), serta posisi uang yang tidak stabil.

Pada beberapa kasus, model mengalami kesalahan klasifikasi karena adanya kemiripan warna dan bentuk antar nominal uang tertentu. Selain itu, pada kondisi pencahayaan yang kurang baik, model cenderung menghasilkan deteksi yang kurang akurat sehingga beberapa objek tidak terdeteksi.

Kesalahan juga ditemukan berupa *false positive*, yaitu ketika beberapa objek pada latar belakang (*background*) terdeteksi sebagai uang meskipun bukan merupakan objek uang yang sebenarnya. Hal ini diduga disebabkan oleh kemiripan pola, warna, atau tekstur tertentu pada latar belakang dengan karakteristik visual uang yang terdapat pada dataset pelatihan. Meskipun demikian, secara keseluruhan kedua model tetap mampu memberikan performa deteksi yang baik pada sebagian besar data pengujian.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa:

1. Sistem pendeteksi dan penghitung uang rupiah otomatis berbasis YOLOv8 berhasil dikembangkan dan mampu mendeteksi beberapa nominal uang dalam satu gambar serta menghitung total nilai uang secara otomatis.
2. Model YOLOv8n dan YOLOv8s menunjukkan performa yang baik dalam mendeteksi dan mengklasifikasikan nominal uang Rupiah.
3. YOLOv8n memiliki keunggulan pada nilai mAP50-95 yang lebih tinggi. Hal ini menunjukkan kemampuan lokalisasi objek (*bounding box*) yang lebih baik pada berbagai

threshold IoU. Namun, berdasarkan hasil pengujian menggunakan beberapa citra uang rupiah, YOLOv8s menghasilkan jumlah kesalahan deteksi yang lebih sedikit dan mampu melakukan klasifikasi nominal uang dengan lebih konsisten.

4. Beberapa kesalahan deteksi masih ditemukan pada kedua model, terutama pada kondisi uang yang bertumpuk (*overlap*), sebagian tertutup (*occlusion*), pencahayaan yang kurang optimal, serta latar belakang yang terdeteksi sebagai uang meskipun bukan merupakan uang sebenarnya.
5. Model YOLOv8s dipilih untuk diimplementasikan pada *website* karena menunjukkan performa deteksi yang lebih konsisten pada pengujian implementasi. *Website* yang dikembangkan mendukung deteksi melalui unggah gambar maupun kamera secara langsung, sehingga dapat digunakan secara lebih interaktif.
6. Hasil penelitian menunjukkan bahwa algoritma YOLOv8 memiliki potensi yang baik untuk diterapkan pada sistem deteksi objek berbasis *computer vision*, khususnya dalam pengembangan sistem pengenalan dan penghitungan uang rupiah otomatis.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, beberapa saran untuk penelitian selanjutnya adalah sebagai berikut:

1. Menambah jumlah dan variasi dataset agar model lebih mampu mengenali berbagai kondisi uang rupiah.
2. Melakukan pengujian pada lebih banyak data dan kondisi nyata untuk memperoleh hasil yang lebih representatif.
3. Mengoptimalkan parameter pelatihan untuk meningkatkan akurasi deteksi model.
4. Membandingkan YOLOv8 dengan algoritma deteksi objek lainnya pada penelitian selanjutnya.
5. Mengembangkan sistem ke platform *mobile* serta menambahkan fitur pendukung, seperti output suara dan deteksi *real-time* yang lebih optimal.

DAFTAR PUSTAKA

- Akbar, N., & Rahman, A. (2025). Pengaruh Image Size pada Penghitung Jumlah Orang Dalam Ruangan Menggunakan Metode YOLOv8. *Jurnal Algoritme*, 5(2), 176–187. <https://doi.org/10.35957/algoritme.v5i2.9035>
- Azhar, K. M., Santoso, I., & Soetrisno, Y. A. A. (2021). Implementasi Deep Learning Menggunakan Metode Convolutional Neural Network dan Algoritma Yolo dalam Sistem Pendeteksi Uang Kertas Rupiah Bagi Penyandang Low Vision. *Transient: Jurnal Ilmiah Teknik Elektro*, 10(3), 502–509. <https://doi.org/10.14710/transient.v10i3.502-509>
- Azhari, M., Situmorang, Z., & Rosnelly, R. (2021). Perbandingan Akurasi, Recall, dan Presisi Klasifikasi pada Algoritma C4.5, Random Forest, SVM dan Naive Bayes. *Jurnal Media Informatika Budidarma*, 5(2), 640–651. <https://doi.org/10.30865/mib.v5i2.2937>
- Hamdani, I. M., Nurhidayat, Karman, A., Adhalia H, N. F., & Julyaningsih, A. H. (2024). Edukasi dan Pelatihan Data Science dan Data Preprocessing. *Intisari: Jurnal Inovasi Pengabdian Masyarakat*, 2(1), 19–26. <https://doi.org/10.58227/intisari.v2i1.125>
- Hapsari, T. R., & Indrati, S. (2022). Efektivitas Penggunaan Website dengan Perhitungan Recall dan Precision. *Indonesian Journal of Social Science Review*, 2(2), 127–135. <https://doi.org/10.26623/jisl.v3i2.5995>
- Iqbal, A. M., Jumadi, J., & Nurlatifah, E. (2025). Implementasi YOLOv8 Sebagai Pendeteksi Nominal Uang Rupiah Kertas Berbasis Android. *SMATIKA JURNAL*, 15(02), 268–277. <https://doi.org/10.32664/smatika.v15i02.1545>
- Marchiningrum, A. U., Akbar Gozali, A., Satria Pradithae, V., & Harry Prayogo, A. (2026). Deteksi Otomatis Kematangan Buah Kelapa Sawit Menggunakan YOLOv8 untuk Pemilihan Model Deteksi Terbaik. *Jurnal Pendidikan dan Teknologi Indonesia*, 6(2), 299–306. <https://doi.org/10.52436/1.jpti.1135>
- Money Detection Dataset. (n.d.). [Dataset Repository]. Roboflow Universe. Retrieved May 20, 2026, from <https://universe.roboflow.com/rupiah-detection/rupiah-detection-vcryt/browse?queryText=&pageSize=50&startingIndex=0&browseQuery=true>
- Pambudi, A. R., Garno, & Purwanto. (2020). Deteksi Keaslian Uang Kertas Berdasarkan Watermark Dengan Pengolahan Citra Digital. *Jurnal Informatika Polinema*, 6(4), 69–74. <https://doi.org/10.33795/jip.v6i4.407>
- Pratama, A. R., Mustajib, M., & Nugroho, A. (2020). Deteksi Citra Uang Kertas dengan Fitur RGB Menggunakan K-Nearest Neighbor. *Jurnal Eksplora Informatika*, 9(2), 163–172. <https://doi.org/10.30864/eksplora.v9i2.336>
- Pratama, M. D. R., Priyatna, B., Hilabi, S. S., & Hananto, A. L. (2022). Deteksi Objek Kecelakaan Pada Kendaraan Roda Empat Menggunakan Algoritma YOLOv5. *Teknologi: Jurnal Ilmiah Sistem Informasi*, 12(2), 15–26. <https://doi.org/10.26594/teknologi.v12i2.3260>

- Rachmawati, F., & Widhyaestoeti, D. (n.d.). Deteksi Jumlah Kendaraan di Jalur SSA Kota Bogor Menggunakan Algoritma Deep Learning YOLO. *Deteksi Jumlah Kendaraan di Jalur SSA Kota Bogor Menggunakan Algoritma Deep Learning YOLO*, 360–370.
- Rafikil, D., Basri, M., Suwardoyo, U., Wafiah, A., & Yunus, M. (2025). Pendeteksi Nominal Mata Uang Kertas menggunakan Kamera berbasis Android. *Jurnal Manajemen Informatika, Sistem Informasi dan Teknologi Komputer (JUMISTIK)*, 4(2), 495–504. <https://doi.org/10.70247/jumistik.v4i2.170>
- Rizkilillah, M. F., Alfarissi, S., & Muharram, M. (2024). Pengembangan Model Klasifikasi Nominal Uang Kertas Rupiah Berbasis Transfer Learning Untuk Tunanetra.
- Steven Immanuel Sihombing, R., Abadi Harahap, W., & Kurnia Rahman, W. (2024). Implementasi Yolo V8 untuk Mendeteksi Mata Uang Rupiah Emisi Tahun 2022 Ber-Output Audio. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(4), 5900–5905. <https://doi.org/10.36040/jati.v8i4.10099>
- Sutisna, T., Raharja, A. R., Solihin, Hariyadi, E., & Putra, V. H. C. (2024). Penggunaan Computer Vision untuk Menghitung Jumlah Kendaraan dengan Menggunakan Metode SSD (SingleShootDetector). *INNOVATIVE: Journal Of Social Science Research*, 4(2), 6060–6067.
- Tanujaya, C. A., Azhar, N. F., & Nugroho, B. (2026). Pengembangan Sistem Anti-Spoofing Berbasis Face Recognition Menggunakan Arsitektur YOLOv8n. *Journal of Applied Computer Science and Technology*, 6(2), 149–161. <https://doi.org/10.52158/jacost.v6i2.1362>
- Taufiqurrahman, Hadi, A. P., & Siregar, R. E. (2024). Evaluasi Performa YOLOv8 Dalam Deteksi Objek Di Depan Kendaraan Dengan Variasi Kondisi Lingkungan. *Jurnal Minfo Polgan*, 13(2), 1755–1773. <https://doi.org/10.33395/jmp.v13i2.14228>
- Upuy, D., Talakua, M. W., Ilwaru, V. Y. I., Balami, A. M., & Hiariey, A. H. (2026). Analisis Komparatif Model YOLOv8n Dan YOLOv8s Untuk Deteksi Objek Berdasarkan Metrik Evaluasi Dan Confidence Score. *Jurnal Computer Science and Information Technology*, 7(1), 51–57. <https://doi.org/10.37859/coscitech.v7i1.11189>
- Wahyuni, S., & Sulaeman, M. (2022). Penerapan Algoritma Deep Learning Untuk Sistem Absensi Kehadiran Deteksi Wajah Di PT Karya Komponen Presisi. *Jurnal Informatika SIMANTIK*, 7(1), 12–21.