

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian yang Relevan

Penelitian terdahulu sangat penting sebagai landasan untuk memahami metode, hasil, serta perkembangan teknologi terkait. Melalui kajian ini, kelebihan, keterbatasan, dan peluang pengembangan dapat diidentifikasi. Penelitian ini mengkaji penerapan algoritma YOLOv5 dan YOLOv8 pada berbagai kasus deteksi objek, termasuk pemilahan sampah. Hasil kajian digunakan untuk menentukan kebaruan (*novelty*) penelitian ini, yaitu deteksi sampah menggunakan YOLOv8 Nano berbasis analisis variasi ukuran objek (*multi-scale*).

Penelitian relevan pertama dijalankan oleh (Faqih Al Firdaus et al., 2025) mengenai penerapan algoritma YOLOv8 untuk mengklasifikasikan sampah ke dalam kategori organik dan *recyclable* demi mendukung otomatisasi sistem pemilahan sampah. Studi ini memanfaatkan *dataset* dari Kaggle berjumlah 698 citra yang terbagi menjadi dua kelas (*organic* dan *recyclable*). Pengolahan data diawali dengan proses anotasi melalui platform Roboflow, yang dilanjutkan dengan pembagian *dataset* menggunakan rasio 70% data latih, 20% data validasi, dan 10% data uji. Arsitektur yang diterapkan adalah YOLOv8m berpixel 640×640 , dengan konfigurasi pelatihan selama 100 *epoch*, *batch size* 16, *learning rate* 0,001, serta optimasi *Stochastic Gradient Descent* (SGD) via Google Colaboratory. Kinerja model diukur berdasarkan metrik *Precision*, *Recall*, *mean Average Precision* (mAP), *F1-Score*, dan *Confusion Matrix*. Evaluasi menunjukkan penurunan konsisten pada nilai *box loss*, *classification loss*, dan *distribution focal loss*, seiring peningkatan nilai mAP serta *Recall* hingga model mencapai titik konvergen. Temuan ini membuktikan tingginya kemampuan generalisasi model dalam memilah sampah organik dan *recyclable*, sehingga berpotensi kuat untuk diimplementasikan pada sistem pemilahan sampah otomatis yang efektif dan efisien.

Selanjutnya, (Riyadi et al., 2024) melakukan studi komparasi antara performa YOLOv5 dan YOLOv8 dalam mendeteksi objek sampah pada fasilitas daur ulang menggunakan *Waste Recycling Plant* (WaRP) *Dataset*. Data yang digunakan

mencakup 2.452 citra untuk pelatihan dan 522 citra untuk validasi beresolusi 1920×1080 piksel, yang terbagi dalam 28 kelas objek dan dikelompokkan ke dalam lima kategori utama (*bottles*, *carton*, *detergent*, *canisters*, dan *cans*). *Dataset* ini merepresentasikan kompleksitas kondisi lingkungan riil, seperti tumpukan antarobjek, variasi pencahayaan, serta pergeseran posisi. Kedua algoritma dilatih selama 100 *epoch* dan dievaluasi menggunakan metrik *Precision*, *Recall*, mAP50, serta mAP50-95. Hasil eksperimen mengonfirmasi keunggulan YOLOv8 atas YOLOv5 di seluruh metrik, dengan perolehan *Precision* 0,569, *Recall* 0,513, mAP50 0,547, dan mAP50-95 0,426 (dibandingkan YOLOv5 yang hanya mencapai *Precision* 0,478, *Recall* 0,442, mAP50 0,406, dan mAP50-95 0,275). Peningkatan ini menunjukkan efektivitas pembaruan arsitektur YOLOv8 dalam menangani deteksi objek pada kondisi visual yang rumit. Kendati demikian, penelitian ini sebatas membandingkan performa umum tanpa menganalisis tingkat akurasi berdasarkan skala ukuran objek (*small*, *medium*, dan *large*), sehingga menyisakan celah untuk eksplorasi lebih lanjut terkait deteksi multiskala.

Penelitian relevan ketiga oleh (Arvio, Kusuma, & BM Sangadji, 2024) berfokus pada pengembangan aplikasi pemilahan sampah anorganik berbasis YOLOv8 guna mendukung *smart computing* dan pemanfaatan sampah sebagai sumber energi terbarukan via metode Tempat Olah Sampah Setempat (TOSS). *Dataset* penelitian ini terdiri dari 2.000 citra dengan lima kategori sampah anorganik: *plastic bottle*, *plastic*, *glass*, *can*, dan *styrofoam*. Model dievaluasi menggunakan metrik *Accuracy*, *Precision*, *Recall*, dan *F1-Score* untuk menguji performa deteksi *real-time*. Hasil pengujian laboratorium mencatatkan rata-rata *accuracy* 99,33%, *precision* 99,63%, *recall* 96,53%, dan *F1-Score* 98,03%, sementara pengujian lapangan secara langsung menghasilkan rata-rata akurasi sebesar 98%. Perolehan ini mengindikasikan keandalan YOLOv8 dalam mengenali variasi jenis sampah anorganik secara otomatis. Meskipun begitu, pengujian yang dilakukan terbatas pada lima kelas anorganik serta belum menyertakan evaluasi berdasarkan variasi ukuran objek (*small*, *medium*, dan *large*). Berdasarkan keterbatasan tersebut, penelitian ini dikembangkan untuk memberikan evaluasi yang lebih komprehensif melalui penerapan varian YOLOv8 Nano serta

analisis performa deteksi pada setiap kategori skala objek guna mengukur ketahanan model terhadap variasi ukuran sampah di kondisi lapangan.

Penelitian relevan keempat oleh (Fahmi et al., 2025) menerapkan algoritma YOLOv8 Nano (YOLOv8n) untuk mendeteksi sampah organik dan anorganik pada sistem *smart waste management*. Studi ini memanfaatkan 3.169 citra dari Roboflow (2.018 beranotasi) dengan *pre-processing* berupa *resizing* 640×640 piksel, normalisasi, augmentasi, serta pembagian data 60:20:20 (*train:val:test*). Pelatihan dilakukan melalui *transfer learning* selama 100 *epoch* (*batch size* 32, *patience* 15). Hasil evaluasi mencatatkan performa yang sangat tinggi dengan *Precision* 0,968, *Recall* 0,970, *mAP@0.5* 0,986, dan *mAP@0.5:0.95* 0,922. Meskipun *confusion matrix* menunjukkan akurasi kelas anorganik sedikit lebih rendah akibat kompleksitas tekstur dan pencahayaan, YOLOv8n terbukti sangat potensial untuk *smart waste management*. Kendati demikian, penelitian tersebut belum menguji performa model berdasarkan kategori skala ukuran objek (*small, medium, large*). Hal ini menjadi landasan bagi penelitian ini untuk mengevaluasi kemampuan deteksi YOLOv8 Nano secara lebih komprehensif pada berbagai variasi skala ukuran sampah di kondisi nyata.

Penelitian relevan selanjutnya dijalankan oleh (Arvio, Kusuma, & Sangadji, 2024) yang mengkaji efektivitas algoritma YOLOv5 dalam mendeteksi cacat masker pada lini produksi untuk mengotomatisasi *quality control*. Studi ini memanfaatkan 1.000 citra dari kamera *smartphone* dengan anotasi *bounding box* untuk dua kategori: masker normal dan masker cacat (tali terputus). Setelah melalui *pre-processing* berupa *resizing* ke 640×640 piksel dan pembagian data (*train, val, test*), pengujian dilakukan dengan membandingkan tiga varian, yaitu YOLOv5n, YOLOv5s, dan YOLOv5m yang dilatih selama 500 *epoch*. Hasil eksperimen memperlihatkan YOLOv5m meraih akurasi tertinggi mencapai 100%, sedangkan YOLOv5n dan YOLOv5s mencatatkan akurasi 96,6%. Namun, dari aspek kecepatan inferensi, YOLOv5n unggul sebagai yang tercepat dengan waktu komputasi 146 ms, disusul YOLOv5s (344 ms) dan YOLOv5m (537 ms). Temuan ini menyimpulkan adanya *trade-off* di mana YOLOv5m unggul pada akurasi, sementara YOLOv5n lebih optimal untuk pengujian *real-time*. Kendati demikian, penelitian tersebut masih berbasis arsitektur YOLOv5 sehingga belum mengadopsi keunggulan YOLOv8, seperti fitur *anchor-free* dan *decoupled head* yang

terbukti meningkatkan presisi deteksi. Oleh sebab itu, penelitian ini menerapkan arsitektur YOLOv8 Nano sebagai bentuk modernisasi model guna mencapai keseimbangan optimal antara akurasi dan efisiensi komputasi.

Penelitian keenam oleh (Suwela & Zidan Hedriyadi, 2025) menerapkan *transfer learning* pada YOLOv8 untuk membedakan jenis sampah secara otomatis. Menggunakan *dataset* citra teranotasi yang melalui tahap *pre-processing*, model berhasil meraih mAP sebesar 0,88 dengan kecepatan *inference* yang tinggi. Penelitian ini menyimpulkan YOLOv8 efektif untuk pendeteksian sampah, namun menyarankan penambahan variasi *dataset* guna meminimalkan bias serta integrasi fitur perhitungan berat dan pelaporan otomatis.

Penelitian ketujuh oleh (Syaifuddin et al., 2026) mengembangkan sistem pemilahan sampah otomatis berbasis *Edge AI* mengintegrasikan YOLOv8 Nano (YOLOv8n) pada Raspberry Pi 5, Camera Module 3, dan *dashboard real-time* berbasis *Server-Sent Events* (SSE). Pelatihan model menggunakan *optimizer* Adam (100 epoch, *learning rate* 0,001, *batch size* 16). Evaluasi mencatatkan nilai *Precision* dan *Recall* stabil di atas 97%, mAP@0.5 sebesar 98,5%, serta mAP@0.5:0.95 mencapai 90%. Pada perangkat *edge*, sistem menghasilkan latensi 0,5–0,6 ms dengan keberhasilan *re-upload* data 99,7%. Meskipun terbukti akurat dan responsif, studi ini sebatas berfokus pada implementasi perangkat tanpa mengevaluasi performa model berdasarkan variasi skala ukuran objek (*small, medium, large*). Keterbatasan tersebut melandasi penelitian ini untuk menganalisis keandalan deteksi YOLOv8 Nano secara spesifik di berbagai skala ukuran sampah.

Penelitian kedelapan oleh (Urlamma et al., 2024) menerapkan algoritma YOLOv8 untuk mengklasifikasikan 12 kategori sampah (*cardboard, paper, metal, green glass, brown glass, white glass, clothes, shoes, biological, plastic, batteries, dan trash*) secara *real-time* berbasis *camera feed*. Dataset yang teranotasi melalui tahap *pre-processing* berupa *resizing*, normalisasi, serta augmentasi data untuk mendukung skema *single-stage detector*. Evaluasi yang divisualisasikan melalui kurva *Precision–Recall* dan *Precision–Confidence* menunjukkan bahwa YOLOv8 mampu mengklasifikasikan ke-12 jenis sampah secara akurat dan responsif untuk sistem otomasi pemilahan. Kendati demikian, penelitian ini belum menyajikan metrik kuantitatif lengkap (seperti nilai

Precision, *Recall*, dan *mAP*) serta belum menganalisis performa berdasarkan variasi skala objek. Keterbatasan tersebut melandasi penelitian ini untuk melengkapi evaluasi menggunakan YOLOv8 Nano berbasis metrik *Precision*, *Recall*, *IoU*, *mAP*, serta analisis performa pada tiap kategori skala objek (*small*, *medium*, dan *large*).

Penelitian kesembilan oleh (Visen & Charibaldi, 2025) mengembangkan sistem deteksi objek menggunakan algoritma YOLOv8 untuk mengenali berbagai jenis sampah anorganik dalam satu citra sebagai upaya mendukung proses pemilahan sampah secara otomatis. Penelitian ini menggunakan 12 kategori sampah, yaitu *cardboard*, *paper*, *metal*, *green glass*, *brown glass*, *white glass*, *clothes*, *shoes*, *biological*, *plastic*, *batteries*, dan *trash*. Sebelum proses pelatihan, seluruh citra melalui tahapan anotasi, *resizing*, normalisasi, serta augmentasi agar sesuai dengan kebutuhan input model YOLOv8. Pelatihan dilakukan selama 100 epoch, sedangkan evaluasi model menggunakan metrik *Precision*, *Recall*, *mAP@0.5* (*mAP50*), dan *mAP@0.5:0.95* (*mAP50-95*). Hasil penelitian menunjukkan bahwa model memperoleh *Precision* sebesar 86,2%, *Recall* sebesar 79,1%, *mAP50* sebesar 87,1%, dan *mAP50-95* sebesar 72,1%, dengan nilai *box loss* sebesar 0,7313 dan *classification loss* sebesar 0,4008. Pengujian lebih lanjut memperlihatkan bahwa model mampu mendeteksi objek sampah secara akurat pada citra yang memuat 1 hingga 10 objek, namun performanya menurun ketika jumlah objek melebihi sepuluh akibat meningkatnya kompleksitas citra, seperti kondisi objek yang saling bertumpang tindih (*occlusion*) dan kepadatan objek yang tinggi. Penelitian tersebut menyimpulkan bahwa algoritma YOLOv8 memiliki kemampuan yang baik dalam mendeteksi berbagai jenis sampah anorganik secara otomatis, tetapi masih memerlukan penambahan jumlah dan variasi dataset agar mampu meningkatkan akurasi pada kondisi citra yang lebih kompleks. Meskipun demikian, penelitian ini belum melakukan analisis performa berdasarkan kategori ukuran objek (*small*, *medium*, dan *large*), sehingga penelitian yang dilakukan mengembangkan evaluasi yang lebih komprehensif melalui analisis deteksi multiskala menggunakan YOLOv8 Nano untuk mengetahui pengaruh variasi ukuran objek terhadap performa model.

Penelitian relevan terakhir oleh (Kestane et al., 2024) mengembangkan sistem deteksi sampah daur ulang secara *real-time* menggunakan YOLOv8 yang

diintegrasikan pada *Reverse Vending Machine* (RVM). Dataset yang diperkaya melalui augmentasi menghasilkan 4.000 citra *resizing* 640×640 piksel untuk tiga kelas sampah (botol plastik/PET, botol kaca, dan kaleng aluminium). Setelah dilatih selama 100 epoch, evaluasi model mencatatkan performa sangat tinggi dengan *Precision* 98,3%, *Recall* 97,8%, *mAP@0.5* sebesar 99,1%, dan *mAP@0.5:0.95* sebesar 87,8%. Hasil ini menyimpulkan keandalan YOLOv8 untuk pemilahan sampah otomatis pada sistem RVM. Kendati demikian, penerapannya masih terbatas pada tiga jenis material serta belum menganalisis performa berdasarkan variasi skala objek (*small*, *medium*, dan *large*). Hal tersebut melandasi penelitian ini untuk melengkapi evaluasi menggunakan YOLOv8 Nano pada sampah organik dan anorganik berbasis analisis multiskala guna mengukur kemampuan deteksi model terhadap variasi ukuran objek di kondisi nyata.

2.1.1 Analisis Gap Penelitian

Kajian terhadap beberapa penelitian terdahulu memperlihatkan perkembangan yang signifikan dalam pemanfaatan *computer vision* untuk otomasi pengelolaan sampah. Penelitian yang dilakukan oleh (Faqih Al Firdaus et al., 2025), (Arvio, Kusuma, & BM Sangadji, 2024), serta (Suwela & Zidan Hedriyadi, 2025) berhasil membuktikan bahwa arsitektur YOLOv8 memiliki tingkat akurasi dan konvergensi yang tinggi dalam mengklasifikasikan berbagai jenis sampah, baik organik maupun anorganik. Selanjutnya, (Riyadi et al., 2024) mengonfirmasi keunggulan arsitektur YOLOv8 dibandingkan pendahulunya (YOLOv5) pada kondisi visual lingkungan nyata yang rumit, seperti tumpukan objek dan variasi pencahayaan. Di sisi lain, (Arvio, Kusuma, & Sangadji, 2024) menunjukkan pentingnya pemilihan varian model; meskipun model berukuran medium seperti YOLOv5m menawarkan akurasi tinggi, varian *nano* (YOLOv5n) terbukti jauh lebih efisien untuk aplikasi *real-time* karena latensi komputasinya yang rendah dan masih menyisakan keterbatasan karena belum mengadopsi pembaruan struktur YOLOv8, seperti pendekatan *anchor-free* dan *decoupled head* yang mampu mengoptimalkan presisi deteksi. Keterbatasan ini dijawab oleh (Fahmi et al., 2025) dan (Syafuddin et al., 2026) yang mengimplementasikan

varian YOLOv8 Nano (YOLOv8n). Kedua penelitian tersebut membuktikan bahwa YOLOv8n merupakan model yang sangat responsif, presisi, dan efisien saat diintegrasikan dengan perangkat *Edge AI*. Kendati demikian, seluruh penelitian terdahulu tersebut memiliki satu benang merah keterbatasan yang belum terselesaikan: pengujian dan evaluasi performa model yang dilakukan masih bersifat umum (*overall metric*) dan berfokus pada integrasi sistem. Belum ada penelitian terdahulu yang menganalisis kemampuan model secara rinci berdasarkan diferensiasi skala ukuran objek.

Ketiadaan analisis skala objek ini menjadi celah penelitian (*research gap*) yang krusial. Pada kondisi riil di lingkungan pembuangan maupun fasilitas daur ulang, citra sampah tidak selalu hadir dalam ukuran yang seragam. Objek sampah kerap muncul dalam dimensi visual yang bervariasi, mulai dari pecahan kecil (*small*), botol atau kaleng berukuran sedang (*medium*), hingga tumpukan kardus atau wadah berukuran besar (*large*). Kinerja algoritma deteksi objek—terutama varian ringan seperti Nano—sangat rentan mengalami penurunan akurasi ketika berhadapan dengan objek berukuran sangat kecil atau objek bertumpuk yang mengaburkan batas skala.

Guna mengisi celah penelitian tersebut, penelitian ini mengajukan penerapan algoritma YOLOv8 Nano (YOLOv8n) yang dikhususkan pada analisis deteksi citra sampah secara *multi-scale*. Kebaruan (*novelty*) dan kontribusi utama penelitian ini terletak pada proses evaluasi komprehensif yang membedah metrik performa model secara terpisah berdasarkan kategori skala ukuran objek (*small*, *medium*, dan *large*). Melalui pendekatan ini, ketahanan (*robustness*) serta batas kemampuan YOLOv8 Nano dalam mendeteksi variasi dimensi sampah di kondisi nyata dapat teruji secara empiris, sekaligus menjaga keseimbangan antara tingkat akurasi deteksi dan efisiensi komputasi.

2.2 Landasan Teori

2.2.1 Konsep Pengelolaan Sampah

Pengolahan sampah merupakan tahap penting dalam manajemen persampahan yang bertujuan untuk mengurangi volume, mengubah karakteristik,

dan memanfaatkan kembali nilai sumber daya dari sampah perkotaan (*Municipal Solid Waste*). Sistem pengolahan konvensional yang berfokus pada pembuangan langsung ke Tempat Pembuangan Akhir (TPA) berisiko tinggi mencemari lingkungan akibat emisi gas rumah kaca (CH_4) dan resapan air lindi. Sebagai solusinya, pendekatan modern menerapkan paradigma *Zero Waste* dan ekonomi sirkular yang menekankan pada pencegahan timbulan sampah (*waste avoidance*), penggunaan kembali (*reuse*), serta daur ulang (*recycling*) guna meminimalkan residu yang dibuang ke lingkungan (Nizar et al., n.d.).

2.2.2 Machine Learning

Machine Learning merupakan salah satu cabang ilmu komputer yang berfokus pada pengembangan sistem dan algoritma yang mampu mempelajari pola dari data sehingga dapat melakukan prediksi atau pengambilan keputusan tanpa harus diprogram secara eksplisit untuk setiap tugas tertentu. Menurut (Qiu et al., 2016) Machine Learning merupakan bidang penelitian yang mempelajari teori, performa, dan karakteristik algoritma pembelajaran, serta mengintegrasikan berbagai disiplin ilmu seperti kecerdasan buatan, statistika, optimasi, teori informasi, dan matematika. Kemampuan tersebut menjadikan Machine Learning sebagai salah satu teknologi utama dalam menyelesaikan berbagai permasalahan yang melibatkan data dalam jumlah besar.

Secara umum, Machine Learning bekerja dengan memanfaatkan data sebagai sumber pembelajaran untuk membangun sebuah model. Model tersebut akan mempelajari hubungan atau pola yang terdapat pada data latih (*training data*) sehingga mampu memberikan prediksi atau klasifikasi terhadap data baru yang belum pernah diproses sebelumnya. Semakin baik kualitas dan jumlah data yang digunakan selama proses pelatihan, maka semakin tinggi pula kemampuan model dalam melakukan prediksi secara akurat.

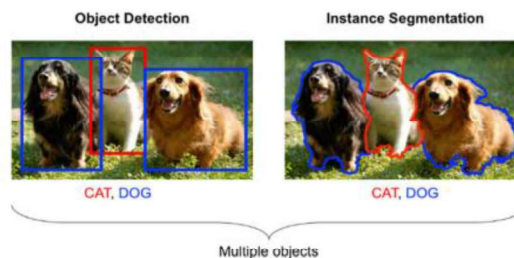
(Qiu et al., 2016) menjelaskan bahwa Machine Learning telah diterapkan pada berbagai bidang, seperti sistem rekomendasi, pengenalan pola (*pattern recognition*), data mining, sistem kendali otomatis, analisis citra, serta pengolahan data skala besar (*big data*). Penerapan tersebut turut mencakup teknik

unsupervised learning seperti algoritma clustering, yang digunakan secara luas untuk mengelompokkan data berdasarkan kemiripan pola tanpa memerlukan label kelas sebelumnya, sebagaimana diterapkan pada analisis profil beban listrik (*electrical load profiling*) (Kusuma et al., 2024).

Dalam penelitian ini, Machine Learning dimanfaatkan untuk membangun model deteksi objek yang mampu mengenali dan mengidentifikasi objek sampah pada citra menggunakan algoritma YOLOv8 Nano. Melalui proses pembelajaran dari dataset yang telah dianotasi, model diharapkan mampu mendeteksi objek sampah secara otomatis dengan tingkat akurasi yang baik pada berbagai ukuran objek (*multi-scale*).

2.2.3 Deteksi Objek (Object Detection)

Deteksi objek merupakan salah satu tugas dalam bidang *computer vision* yang bertujuan untuk mengidentifikasi objek dalam citra sekaligus menentukan lokasinya. Berbeda dengan klasifikasi citra, deteksi objek tidak hanya mengenali jenis objek, tetapi juga memberikan informasi posisi dalam bentuk *bounding box* dan label kelas. Teknologi ini banyak digunakan dalam berbagai bidang, seperti sistem keamanan, kendaraan otonom, dan pengolahan citra lingkungan. Dalam penelitian ini, deteksi objek digunakan untuk mengenali dan menentukan lokasi objek sampah secara otomatis dan dalam bentuk multi-scale



Gambar 2. 1 Object Detection dan Instance Segmentation(Diwan et al., 2023)

2.2.4 Multi Skala

Pendekatan multiskala bertujuan memanfaatkan informasi fitur pada berbagai tingkat resolusi sehingga objek dengan ukuran kecil, sedang, maupun besar dapat direpresentasikan secara lebih optimal. Integrasi informasi dari beberapa skala terbukti mampu meningkatkan kemampuan deteksi objek pada

kondisi lingkungan yang beragam. Penerapan adaptasi domain pada seluruh skala fitur menghasilkan performa yang lebih baik dibandingkan hanya menerapkannya pada satu atau dua skala. Hal ini menunjukkan bahwa pembelajaran multiskala mampu meningkatkan kualitas representasi fitur dari lapisan awal hingga akhir jaringan (Hnewa & Radha, 2022).

2.2.4.1 Permasalahan Objek dengan Ukuran Berbeda

Variasi ukuran objek merupakan salah satu tantangan utama dalam tugas deteksi objek. Dalam sebuah citra, objek dapat muncul dengan ukuran yang sangat kecil, sedang, maupun besar akibat perbedaan jarak objek terhadap kamera, sudut pengambilan gambar, maupun resolusi citra. Kondisi tersebut menyebabkan model deteksi objek harus mampu mengenali karakteristik objek pada berbagai tingkat skala tanpa mengurangi akurasi lokalisasi maupun klasifikasi (Elizar et al., 2022).

2.2.4.2 Dampak Variasi Skala terhadap Performa Model

Variasi skala objek memberikan pengaruh yang signifikan terhadap performa model deteksi objek. Objek yang memiliki ukuran sangat kecil cenderung kehilangan informasi penting selama proses ekstraksi fitur, sedangkan objek yang berukuran besar membutuhkan cakupan konteks yang lebih luas agar dapat dikenali secara optimal. Kondisi ini menyebabkan model deteksi objek mengalami kesulitan dalam mempertahankan akurasi ketika objek muncul dalam ukuran yang beragam pada satu citra (Elizar et al., 2022).

Pembahasan oleh (Hnewa & Radha, 2022) juga menunjukkan bahwa penggunaan fitur pada satu tingkat skala belum mampu menghasilkan representasi yang optimal. Ketika proses adaptasi hanya diterapkan pada skala fitur terakhir, informasi dari lapisan awal tidak memperoleh pembelajaran yang memadai akibat melemahnya aliran gradien. Dampak dari kondisi tersebut adalah menurunnya kemampuan model dalam mendeteksi objek dengan ukuran yang bervariasi.

Oleh karena itu, model deteksi objek modern seperti YOLOv8 Nano memanfaatkan ekstraksi dan penggabungan fitur multiskala pada beberapa tingkat resolusi untuk meningkatkan kemampuan deteksi terhadap objek kecil, sedang, maupun besar. Dengan memanfaatkan informasi dari berbagai skala fitur, model mampu menghasilkan representasi yang lebih kaya sehingga performa deteksi menjadi lebih stabil pada objek dengan ukuran yang beragam.

2.2.5 Data Augmentation

Data Augmentation (augmentasi data) merupakan teknik manipulasi data yang bertujuan untuk meningkatkan jumlah dan variabilitas data pelatihan secara artifisial tanpa perlu mengumpulkan data baru secara manual. Dalam *Deep Learning*, teknik ini digunakan untuk mengatasi permasalahan *overfitting* akibat terbatasnya kumpulan data latih. Teknik *data augmentation* dapat dikategorikan menjadi *data warping*—seperti transformasi geometris (rotasi, *flipping*, pemotongan), manipulasi ruang warna, dan *random erasing*—serta pendekatan berbasis *deep learning* seperti *Generative Adversarial Networks* (GAN), *Neural Style Transfer*, dan *meta-learning*. Dengan memperluas keragaman sampel masukan, augmentasi membantu model mempelajari fitur yang lebih robust serta meningkatkan kemampuan generalisasi saat melakukan prediksi pada data baru (Shorten & Khoshgoftaar, 2019)

2.2.6 Algoritma Yolo

YOLO (*You Only Look Once*) merupakan kerangka kerja deteksi objek berbasis *deep learning* yang beroperasi secara *real-time* dengan pendekatan *single-shot* atau *end-to-end*. Berbeda dengan metode berbasis dua tahap seperti keluarga R-CNN yang memisahkan proses usulan wilayah (*region proposal*) dan klasifikasi, YOLO membagi citra menjadi kisi-kisi (*grid*) $S \times S$ dan memprediksi koordinat lokasi (*bounding box*) serta probabilitas kelas objek secara bersamaan dalam satu kali eksekusi (*single pass*) melalui jaringan saraf konvolusi (Terven et al., 2023a)



Gambar 2. 2 Algoritma YOLO

Sejak pertama kali diperkenalkan oleh Joseph Redmon et al. pada tahun 2015 melalui YOLOv1, algoritma ini terus mengalami evolusi yang signifikan. Perkembangan dilanjutkan oleh YOLOv2 (YOLO9000) yang mengenalkan *anchor boxes* dan *batch normalization*, serta YOLOv3 yang mengadopsi *backbone* Darknet-53 dengan fitur prediksi multi-skala (*multi-scale predictions*). Kemudian, YOLOv4 mengintegrasikan kombinasi teknik *Bag-of-Freebies* dan *Bag-of-Specials*, yang diikuti oleh YOLOv5 besutan Ultralytics dengan optimasi PyTorch dan fitur *AutoAnchor*. Inovasi berlanjut pada YOLOv6 dan YOLOv7 yang mengoptimalkan arsitektur *re-parameterization* serta *E-ELAN*, hingga berlanjut pada YOLOv8 yang menerapkan pendekatan *anchor-free* dan *decoupled head* untuk mendukung berbagai tugas vision seperti deteksi, segmentasi, hingga estimasi pose (Terven et al., 2023a)

Keunggulan utama YOLO dibandingkan algoritma deteksi objek lainnya terletak pada kecepatan inferensinya yang sangat tinggi sehingga ideal untuk pemrosesan *real-time* pada sistem otonom dan pemantauan video. Selain itu, karena YOLO melihat keseluruhan citra secara global saat melakukan prediksi, algoritma ini mampu meminimalkan tingkat kesalahan prediksi latar belakang (*background false positives*). YOLO juga menawarkan fleksibilitas skabilitas model (mulai dari varian *nano* hingga *extra-large*), yang memberikan keseimbangan optimal antara kecepatan dan akurasi untuk disesuaikan dengan kapasitas perangkat keras (*hardware*)(Terven et al., 2023a)

1. Representasi Citra *Input* (RGB)

Citra masukan pada algoritma YOLO direpresentasikan sebagai tensor tiga dimensi dengan dimensi tinggi (H), lebar (W), dan jumlah saluran warna (C). Untuk penelitian ini, citra dinormalisasi ke ukuran 640×640 piksel dengan 3 saluran warna (RGB), sehingga direpresentasikan sebagai:

$$X \in RH \times W \times C$$

Di mana X adalah matriks citra, $H=640$, $W=640$, dan $C=3$. Setiap nilai piksel dinormalisasi ke rentang $[0, 1]$ untuk mempercepat konvergensi model.

2. Operasi Konvolusi

Ekstraksi fitur pada *backbone* CSPDarknet dilakukan menggunakan operasi konvolusi 2D. Operasi ini menghitung hasil kali titik antara *kernel* (bobot) dan area lokal pada citra input, yang dirumuskan sebagai:

$$y_{i,j} = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} x_{i+m,j+n} \cdot w_{m,n} + b$$

Di mana $y_{i,j}$ adalah nilai output pada posisi (i,j) , x adalah nilai input, w adalah bobot *kernel* berukuran $k \times k$, dan b adalah nilai *bias*.

3. Intersection over Union (IoU)

Intersection over Union (IoU) digunakan untuk mengukur tingkat tumpang tindih antara *bounding box* prediksi (B_{pred}) dengan *bounding box ground truth* (B_{gt}). Nilai IoU menjadi dasar evaluasi akurasi lokalisasi objek:

$$IoU = \frac{Area(B_{pred} \cap B_{gt})}{Area(B_{pred} \cup B_{gt})}$$

4. Confidences Score

Confidence score merepresentasikan tingkat kepercayaan model terhadap keberadaan objek di dalam sebuah *bounding box*. Secara matematis, *confidence score* didefinisikan sebagai:

$$Confidence = P(Object) \times IoU_{pred}^{gt}$$

Di mana $P(Object)$ adalah probabilitas keberadaan objek, dan IoU_{pred}^{gt} adalah nilai IoU antara prediksi dan *ground truth*.

5. Fungsi Kerugian (*Loss Function*)

YOLOv8 menggunakan tiga komponen utama *loss function* untuk mengoptimalkan model selama pelatihan:

a. Complete Intersection over Union (CIoU) Loss

Digunakan untuk optimasi regresi *bounding box* dengan mempertimbangkan area tumpang tindih, jarak pusat, dan rasio aspek:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

b. Binary Cross Entropy (BCE) Loss

Digunakan untuk tugas klasifikasi kelas objek:

$$L_{cls} = - \sum_{i=1}^N [y_i \log(p_i) + (1-y_i) \log(1-p_i)]$$

c. Distribution Focal Loss (DFL)

Digunakan untuk mempercepat konvergensi regresi *bounding box* pada pendekatan *anchor-free*:

$$L_{DFL} = -((y_i + 1 - y) \log(p_i) + (y - y_i) \log(p_i + 1))$$

Total fungsi kerugian (*Total Loss*) pada YOLOv8 merupakan penjumlahan terbobot dari ketiga komponen di atas:

$$L_{total} = \lambda_{box} L_{CIoU} + \lambda_{cls} L_{cls} + \lambda_{dfl} L_{DFL}$$

Di mana λ_{box} , λ_{cls} , dan λ_{dfl} adalah bobot hiperparameter yang mengatur kontribusi masing-masing *loss* terhadap pembaruan gradien.

2.2.7 YOLOv8 Nano

YOLOv8 Nano (YOLOv8n) merupakan varian model terkecil dan paling ringkas dalam keluarga arsitektur YOLOv8 yang dirancang khusus untuk pemrosesan deteksi objek secara *real-time* berbasis pendekatan *anchor-free* (*center point detection*), *backbone* CSPNet dengan modul C2f, serta *decoupled head*. Pemilihan YOLOv8 Nano didasari oleh efisiensi komputasi dan

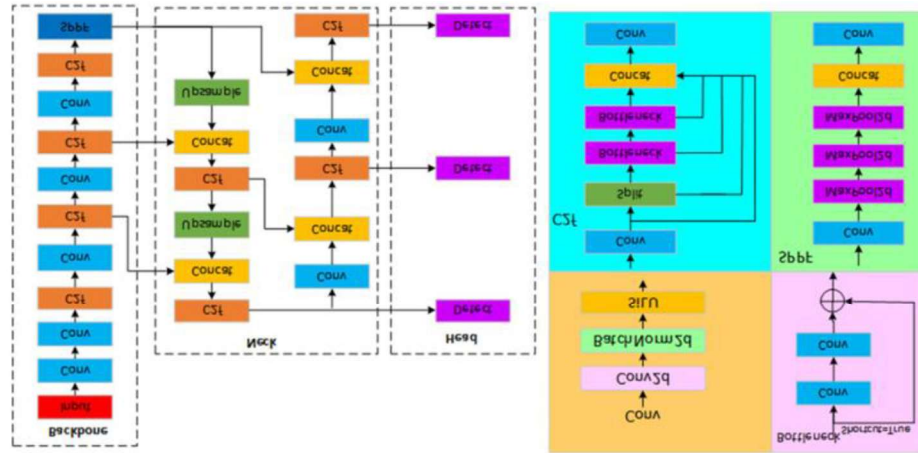
kecepatannya yang sangat tinggi pada perangkat dengan sumber daya terbatas (*edge devices/resource-constrained environments*), di mana model ini mampu mencapai waktu inferensi rata-rata 12 milidetik per *frame* (sekitar 68 FPS) sehingga sangat ideal untuk aplikasi dinamis seperti *Intelligent Transportation Systems* (ITS) maupun pemantauan udara. Keunggulan utama model ini terletak pada penggunaan memori yang minimal dan kinerja yang efisien tanpa mengorbankan akurasi dasar (mencapai tingkat akurasi di atas 80% pada kondisi pencahayaan memadai).

Meskipun demikian, sebagai dampak dari ukurannya yang amat kompak, YOLOv8 Nano memiliki keterbatasan dalam mendeteksi objek berukuran sangat kecil (seperti sepeda motor atau sepeda dari ketinggian tinggi), objek yang terhalang (*occlusion*), serta citra dalam kondisi pencahayaan yang sangat rendah (*low-light*) atau latar belakang yang kompleks (Prayogi & Putra, 2025).

2.2.8 Arsitektur YOLOv8 Nano

YOLOv8 Nano (YOLOv8n) merupakan varian model terkecil dan paling ringkas dalam keluarga arsitektur YOLOv8 yang dirancang khusus untuk pemrosesan deteksi objek secara *real-time* berbasis pendekatan *anchor-free* (*center point detection*), *backbone* CSPNet dengan modul C2f, serta *decoupled head*. Pemilihan YOLOv8 Nano didasari oleh efisiensi komputasi dan kecepatannya yang sangat tinggi pada perangkat dengan sumber daya terbatas (*edge devices/resource-constrained environments*), di mana model ini mampu mencapai waktu inferensi rata-rata 12 milidetik per *frame* (sekitar 68 FPS) sehingga sangat ideal untuk aplikasi dinamis seperti *Intelligent Transportation Systems* (ITS) maupun pemantauan udara. Keunggulan utama model ini terletak pada penggunaan memori yang minimal dan kinerja yang efisien tanpa mengorbankan akurasi dasar (mencapai tingkat akurasi di atas 80% pada kondisi pencahayaan memadai). Meskipun demikian, sebagai dampak dari ukurannya yang amat kompak, YOLOv8 Nano memiliki keterbatasan dalam mendeteksi objek berukuran sangat kecil (seperti sepeda motor atau sepeda dari ketinggian tinggi), objek yang terhalang (*occlusion*), serta citra dalam kondisi pencahayaan

yang sangat rendah (*low-light*) atau latar belakang yang kompleks (Terven et al., 2023b)



Gambar 2. 3 Arsitektur YOLOv8 Nano (YOLOv8n)(Chen et al., 2025)

Gambar 2.3 mengilustrasikan arsitektur YOLOv8 Nano yang terdiri atas tiga komponen utama, yaitu backbone, neck, dan detection head. Pada bagian backbone, proses dimulai dari Input Image yang melewati Convolution-Downsampling Block dan modul C2f untuk mengekstraksi fitur serta mengurangi dimensi spasial secara bertahap. Selanjutnya, fitur diteruskan ke bagian neck yang menggunakan struktur PANet/FPN untuk menggabungkan fitur pada beberapa tingkat resolusi sehingga mendukung deteksi objek dengan berbagai ukuran. Pada bagian akhir, detection head menggunakan pendekatan anchor-free dan decoupled head untuk menghasilkan prediksi berupa koordinat bounding box dan klasifikasi sembilan kelas sampah.(Terven et al., 2023b)

2.2.9 Confusion Matrix

Confusion matrix adalah metode efisien untuk mengevaluasi kinerja klasifikasi melalui tabel yang menunjukkan jumlah data uji benar atau salah, memudahkan identifikasi kesalahan dan penilaian akurasi. Teknik ini sederhana, efektif, dan bertujuan menilai performa sistem dalam mengklasifikasikan data uji (Nurhidayat & Dewi, 2023).

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Gambar 2. 4 Struktur Dasar Confusion Matrix (Confusion Matrix, n.d.)

Keterangan :

- TP (True Positive) = Jumlah data positif yang diklasifikasikan oleh sistem dengan benar.
- TN (True Negative) = Jumlah data negatif yang diklasifikasikan oleh sistem dengan benar.
- FP (False Positive) = Data yang sebenarnya negatif tetapi diprediksi positif.
- FN (False Negative) = Data yang sebenarnya positif tetapi diprediksi negatif.

Confusion matrix digunakan untuk mengevaluasi tingkat akurasi dari suatu proses klasifikasi yang telah dilakukan. Akurasi ini menggambarkan proporsi prediksi yang benar. Untuk menghitung akurasi, presisi, dan recall, diterapkan rumus-rumus yang relevan sebagai berikut:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.1)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2.2)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2.3)$$

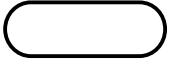
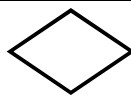
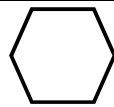
$$\text{F1 - Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.4)$$

2.2.10 FlowChart

Flowchart adalah diagram alir yang dimanfaatkan untuk menggambarkan urutan proses atau logika dalam sebuah system secara visual. Dalam konteks penelitian berbasis kecerdasan buatan, Flowchart berperan dalam

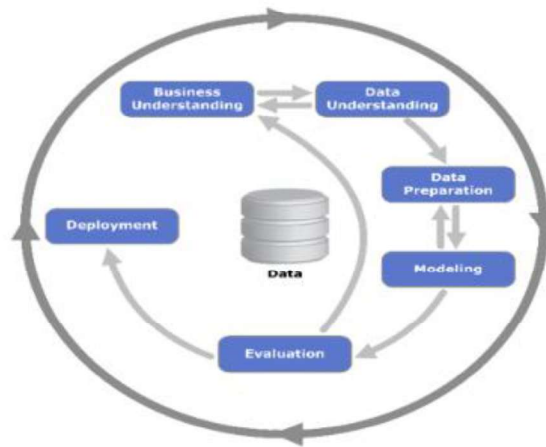
mengilustrasikan tahapan kerja sistem, mulai dari penumpulan data, pra-pemrosesan, pelatihan model evaluasi, hingga interpretasi hasil. Setiap Langkah direpresentasikan dengan symbol standarseperti oval (mulai/akhir), persegi panjang (proses), belah ketupat (keputusan), dan panah (arah aliran data) (Sutanti et al., 2020)

Tabel 2. 1 Flowchart

No	Simbol	Nama	Keterangan
1		Flow Direction	Digunakan untuk menghubungkan simbol yang satu dengan simbol yang lainnya
2		Terminator	Untuk permulaan (start) dan akhir (stop) dari suatu kegiatan
3		Processing	Simbol yang menunjukkan bahwa komputer melakukan pengolahan
4		Decision	Simbol proses yang dipilih berdasarkan kondisi saat ini
5		Connector	Simbol penyambungan proses dalam lembar / halaman yang sama
6		Preparation	Simbol untuk menyiapkan penyimpanan yang akan diproses di dalam penyimpanan
7		Input-Output	Simbol yang menunjukkan proses input dan output terlepas dari jenis peralatan
8		Predefine Proses	Simbol yang menunjukkan pelaksanaan komponen atau prosedur tertentu
9		Display	Simbol yang menunjukkan peralatan output yang digunakan, seperti printer, plotter, layar, dan sebagainya

No	Simbol	Nama	Keterangan
10		Punch Card	Simbol yang menunjukkan bahwa output ditulis ke kartu atau input berasal dari kartu

2.2.11 Kerangka CRISP-DM



Gambar 2. 5 Disain Penelitian dengan CRISP-DM (Siswipraptini et al., 2023)

CRISP-DM (*Cross-Industry Standard Process for Data Mining*) adalah metodologi standar untuk proyek *data mining* dan *machine learning*. Penelitian ini mengacu pada kerangka CRISP-DM dengan enam tahapan sistematis :

1. Business Understanding (Pemahaman Bisnis)

Tahap awal ini bertujuan untuk memahami masalah pemilahan sampah manual yang tidak efisien. Berdasarkan studi literatur, ditetapkan tujuan penelitian yaitu membangun model deteksi 9 kelas sampah *multi-scale* menggunakan YOLOv8.

2. Data Understanding (Pemahaman Data)

Pada tahap ini, dilakukan pengumpulan dan eksplorasi dataset citra sampah dari Kaggle yang berisi 1281 gambar dengan 9 kelas spesifik. Aktivitas meliputi: memeriksa distribusi kelas, menganalisis ukuran citra,

mengidentifikasi variasi ukuran objek, dan memastikan anotasi *bounding box* lengkap.

3. Data Preparation (Persiapan Data)

Tahap persiapan data meliputi *preprocessing* dan augmentasi citra:

- a. Preprocessing: *Resize* ke 640×640 pixel, normalisasi (0-1), konversi warna RGB.
- b. Augmentasi: *Rotation* (0-35°), *width/height shift* (0-25%), *zoom range* (0.5-1.5x), *horizontal flip* (dilakukan via Roboflow).
- c. Split data: 80% training (1024 citra), 10% validation (128 citra), 10% testing (129 citra).

4. Modeling (Pemodelan)

Pada tahap ini, dibangun model deteksi objek menggunakan arsitektur YOLOv8n:

- a. Backbone: CSPDarknet *pre-trained* (COCO).
- b. Neck: PANet (*Path Aggregation Network*) untuk *multi-scale feature fusion*.
- c. *Detection Head*: *Bounding box regression* + *class prediction* (9 kelas).
- d. Training: 100 *epochs*, *batch size* 16, *optimizer* SGD, *learning rate* 0.01.

5. Evaluation (Evaluasi)

Model dievaluasi menggunakan metrik standar deteksi objek:

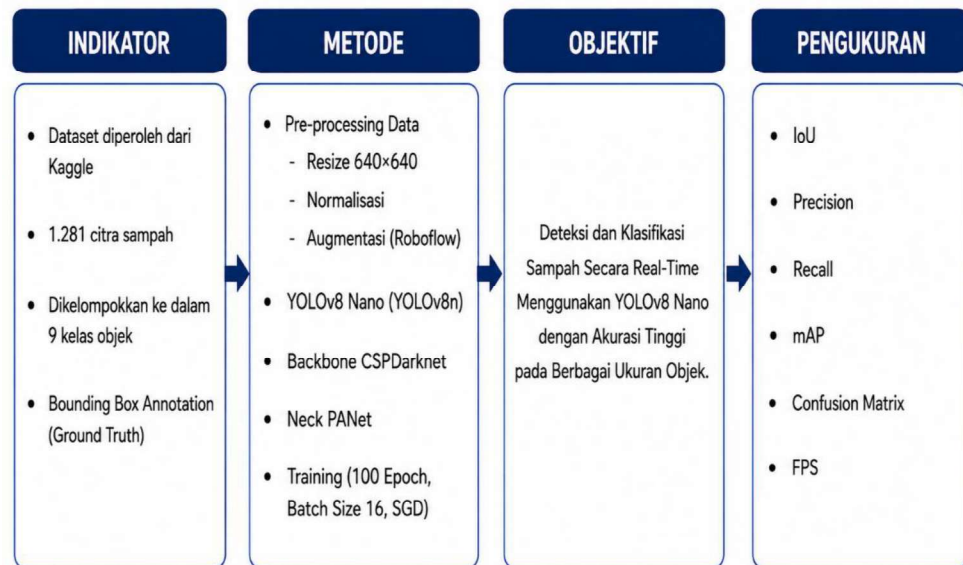
- a. *Intersection over Union* (IoU): Mengukur tumpang tindih *bounding box*.
- b. *mean Average Precision* (mAP): Rata-rata AP untuk semua 9 kelas.
- c. *Multi-Scale Analysis*: Performa untuk *small*, *medium*, *large* objects.

6. Deployment (Implementasi)

Tahap *deployment* pada penelitian ini diwujudkan melalui implementasi prototipe (*proof-of-concept*) sistem deteksi *real-time* menggunakan webcam. Model terbaik (best.pt) yang telah melalui proses evaluasi diintegrasikan ke dalam aplikasi berbasis Python menggunakan pustaka Ultralytics YOLO dan OpenCV. Sistem bekerja dengan menangkap *frame* video secara berurutan, melakukan inferensi model, dan memvisualisasikan hasil deteksi berupa *bounding box*, label kelas, serta *confidence score* secara langsung pada layar.

Tahap ini bertujuan untuk memvalidasi kecepatan inferensi (*Frames Per Second/FPS*) dan membuktikan kemampuan model dalam mendeteksi objek sampah secara *real-time* pada perangkat standar. Hasil dari tahap *deployment* prototipe ini selanjutnya dapat menjadi acuan dasar bagi penelitian selanjutnya untuk melakukan implementasi penuh pada perangkat *edge computing* (seperti Raspberry Pi atau NVIDIA Jetson Nano) di lingkungan produksi seperti Tempat Pembuangan Akhir (TPA).

2.3 Kerangka Pemikiran



Gambar 2. 6 Kerangka Pemikiran

Untuk memberikan pemahaman yang lebih jelas terhadap setiap tahapan dalam kerangka pemikiran di atas, berikut dijelaskan secara rinci masing-masing tahap:

1. Indikator

Tahap pertama adalah indikator, yaitu sumber data yang digunakan dalam penelitian. Dataset diperoleh dari Kaggle dan terdiri atas 1.281 citra sampah yang dikelompokkan ke dalam 9 kelas objek, yaitu *cans_all_type*, *ceramic_product*, *diapers*, *glass_containers*, *paper_products*, *plastic_bags_wrappers*, *plastic_bottles*, *sanitary_napkin*, dan

stroform_product. Seluruh citra telah diberi anotasi menggunakan *bounding box* sebagai *ground truth*, sehingga setiap objek sampah memiliki informasi lokasi dan label kelas yang jelas. Dataset ini menjadi dasar dalam proses pelatihan, validasi, dan pengujian model deteksi objek.

2. Metode

Tahap berikutnya adalah metode, yaitu proses pengolahan data hingga pembentukan model deteksi objek. Sebelum digunakan pada proses pelatihan, dataset terlebih dahulu melalui tahap *pre-processing* yang meliputi *resize* citra menjadi ukuran 640×640 piksel, normalisasi, serta augmentasi data menggunakan Roboflow untuk meningkatkan variasi data dan mengurangi risiko *overfitting*. Selanjutnya, proses deteksi objek dilakukan menggunakan arsitektur YOLOv8 Nano (YOLOv8n) yang memanfaatkan backbone CSPDarknet untuk mengekstraksi fitur penting dari citra serta neck PANet untuk menggabungkan fitur dari berbagai skala sehingga model mampu mendeteksi objek dengan ukuran yang beragam. Model kemudian dilatih menggunakan Google Colab dengan parameter 100 *epoch*, *batch size* 16, dan *optimizer* SGD.

3. Objektif

Tahap objektif merupakan tujuan utama penelitian, yaitu membangun sistem yang mampu mendeteksi dan mengklasifikasikan objek sampah secara otomatis dan *real-time* menggunakan algoritma YOLOv8 Nano (YOLOv8n). Sistem diharapkan mampu mengenali berbagai kategori sampah dengan tingkat akurasi yang baik, termasuk objek yang memiliki ukuran kecil, sedang, maupun besar, sehingga dapat mendukung proses pengelolaan dan pemilahan sampah secara lebih efektif.

4. Pengukuran

Tahap terakhir adalah pengukuran, yaitu proses evaluasi terhadap performa model yang telah dilatih. Evaluasi dilakukan menggunakan beberapa metrik, yaitu:

- a. Intersection over Union (IoU) untuk mengukur tingkat kesesuaian antara bounding box hasil prediksi dengan bounding box sebenarnya (ground truth).
- b. Precision untuk mengukur tingkat ketepatan model dalam mendeteksi objek berdasarkan perbandingan antara prediksi yang benar dengan seluruh hasil prediksi.
- c. Recall untuk mengukur kemampuan model dalam mendeteksi seluruh objek yang terdapat pada data uji.
- d. Mean Average Precision (mAP) sebagai indikator utama dalam mengevaluasi performa keseluruhan model deteksi objek berdasarkan rata-rata nilai Average Precision pada setiap kelas.
- e. Confusion Matrix untuk menganalisis hasil klasifikasi model pada setiap kategori objek sampah, sehingga dapat diketahui jumlah prediksi yang benar maupun kesalahan klasifikasi yang terjadi.
- f. *Frames Per Second* (FPS) untuk mengukur kecepatan inferensi model dalam melakukan deteksi objek secara *real-time* menggunakan *webcam*, sehingga dapat diketahui kemampuan sistem dalam memproses citra secara langsung.