BAB II

TINJAUAN LITERATUR

2.1 Tinjauan Pustaka

Penelitian mengenai *Object Detection* dan *Object Counting* berbasis *Computer Vision* telah banyak dilakukan dengan memanfaatkan algoritma *You Only Look Once* (YOLO). Berbagai penelitian menunjukkan bahwa YOLO mampu mendeteksi objek secara cepat dan dapat diterapkan pada sistem *real-time.* Penggunaan YOLOv8 juga mulai banyak dikembangkan karena memiliki kemampuan deteksi yang baik serta dapat digunakan untuk berbagai jenis objek, seperti kendaraan, manusia, botol, dan objek lainnya. Penelitian mengenai *Object Tracking* Menggunakan YOLOv8 untuk Menghitung Kendaraan menunjukkan bahwa YOLOv8 dapat digunakan untuk mendeteksi dan menghitung objek secara otomatis dengan proses yang cepat dan stabil. Namun, penelitian tersebut masih berfokus pada objek kendaraan (Hayati dkk., 2023).

Selain pada kendaraan, YOLOv8 juga telah digunakan untuk sistem penghitung manusia secara *real-time.* Penelitian mengenai Implementasi YOLOv8 dalam Penghitung Masuk dan Keluar Manusia menunjukkan bahwa YOLOv8 mampu melakukan proses deteksi dan penghitungan objek manusia secara otomatis (Fahrezi dan Widiyanto, 2024). Penelitian lain mengenai Penghitung Jumlah Orang Menggunakan YOLOv8 juga menunjukkan bahwa

YOLOv8 dapat digunakan untuk melakukan *Object Counting* secara *real-time* dengan hasil yang baik (Akbar dan Rahman, 2025). Meskipun demikian, penelitian tersebut masih menggunakan manusia sebagai objek sehingga belum secara khusus diterapkan pada objek farmasi.

Penerapan YOLOv8 juga telah dilakukan pada objek lain, seperti botol dan berbagai objek dalam penelitian perilaku. Penelitian Pengembangan Sistem Deteksi Objek Botol *Real-time* dengan YOLOv8 menunjukkan bahwa YOLOv8 mampu melakukan deteksi objek secara *real-time* dengan tingkat akurasi yang baik (Triyanto dkk., 2024). Sementara itu, penelitian *Automatic Object Detection for Behavioural Research Using* YOLOv8 menunjukkan bahwa YOLOv8 dapat digunakan untuk mendeteksi berbagai objek dalam kondisi penelitian yang berbeda. Namun, penelitian tersebut lebih berfokus pada proses deteksi dan belum secara khusus mengembangkan sistem *Object Counting* pada bidang farmasi (Hermens, 2024).

Penelitian mengenai perkembangan algoritma YOLO juga menunjukkan bahwa metode ini terus mengalami perkembangan dari YOLOv4, YOLOv5 hingga YOLOv8. Penelitian *Research Overview* of YOLO Series *Object Detection Algorithms*, YOLOv8 for *Object Detection: A Comprehensive Review*, dan *Object Detection Using* YOLOv8: *A Systematic Review* membahas perkembangan serta penerapan YOLO dalam berbagai kebutuhan *Object Detection*. Hasil kajian tersebut menunjukkan bahwa YOLOv8 memiliki potensi untuk digunakan dalam sistem deteksi objek secara *real-time.* Namun, penelitian

tersebut sebagian besar berupa kajian literatur sehingga tidak secara langsung menghasilkan sistem *Object Counting* pada objek tertentu (Chen, 2024).

Selain itu, penelitian mengenai Pengaruh *Image Size* pada Penghitung Jumlah Orang Menggunakan YOLOv8 menunjukkan bahwa parameter *Image Size* dapat memengaruhi akurasi hasil deteksi dan penghitungan objek. Hal tersebut menunjukkan bahwa kondisi dan konfigurasi data masukan merupakan salah satu faktor yang perlu diperhatikan dalam pengembangan sistem *Object Counting* (Akbar dan Rahman, 2025). Penelitian lain seperti *Detection Accuracy in Drone Surveillance System* juga menunjukkan bahwa kondisi lingkungan, khususnya pencahayaan, dapat memengaruhi hasil deteksi objek. Oleh karena itu, pengujian sistem pada kondisi yang berbeda diperlukan untuk mengetahui kemampuan dan keterbatasan model (Ayuningtyas, dkk 2025).

Tabel 2. 1Penelitian Terdahulu

No.	Tahun	Judul Penelitian	Metode	Hasil Penelitian	Kelebihan	Kelemahan	Research Gap
1	2023	Object *Tracking* Menggunakan YOLOv8 untuk Menghitung Kendaraan	YOLOv8	Sistem mampu menghitung kendaraan otomatis dengan akurasi tinggi	Deteksi cepat dan stabil	Objek hanya kendaraan	Menghitung objek obat
2	2024	Pengembangan Sistem Deteksi Objek Botol *Real-time* dengan YOLOv8	YOLOv8	Deteksi botol berjalan *real-time*	Akurasi tinggi	Tidak melakukan *Object Counting*	Menggabungkan deteksi dan penghitungan
3	2024	Implementasi YOLOv8	YOLOv8	Sistem menghitung manusia	*Real-time*	Objek manusia	Objek farmasi

		dalam Penghitung Masuk dan Keluar Manusia		otomatis			
4	2024	Automatic *Object Detection* for Behavioural Research Using YOLOv8	YOLOv8	YOLOv8 efektif pada berbagai objek	Akurasi tinggi	Tidak fokus pada industri	Difokuskan pada farmasi
5	2024	Research Overview of YOLO Series *Object Detection* Algorithms	Literature Review	Menjelaskan perkembangan YOLO	Kajian lengkap	Tidak membangun sistem	Menghasilkan implementasi sistem
6	2025	YOLOv8 for *Object Detection*: A Comprehensive Review	Systematic Review	Menjelaskan perkembangan YOLOv8	Referensi komprehensif	Tidak melakukan implementasi	Membangun aplikasi nyata
7	2025	*Object Detection* Using YOLOv8: A Systematic Review	Systematic Review	Menjelaskan perkembangan YOLOv8	Analisis mendalam	Tidak membahas *Object Counting*	Berfokus pada *Object Counting*
8	2025	Pengaruh *Image Size* pada Penghitung Jumlah Orang Menggunakan YOLOv8	YOLOv8	*Image Size* memengaruhi akurasi	Evaluasi parameter	Objek manusia	Objek obat
9	2025	Detection Accuracy in Drone Surveillance System	YOLOv5	Akurasi dipengaruhi pencahayaan	Menguji banyak kondisi	Menggunakan YOLOv5	Menggunakan YOLOv8
10	2026	Sistem Deteksi Otomatis Buah Berbasis	YOLO	Deteksi buah berhasil	Implementasi baik	Tidak menghitung objek	Mengembangkan *Counti*

		Computer Vision Menggunakan YOLO					*ng*
11	2024	Sistem Deteksi Pelanggaran Helm Menggunakan YOLOv5	YOLOv5	Mampu mendeteksi pelanggaran helm	Cocok untuk pengawasan	Tidak menghitung objek	YOLOv8 dan *Object Counting*
12	2022	Deteksi Kendaraan *Real-time* Menggunakan YOLOv5	YOLOv5	Deteksi kendaraan berjalan baik	Akurasi tinggi	Belum menggunakan YOLOv8	Memanfaatkan YOLOv8
13	2024	Vehicle *Counting* Detection Model Using YOLOv4	YOLOv4	Penghitungan kendaraan berhasil	Fokus pada *counting*	Akurasi lebih rendah dibanding YOLOv8	Menggunakan YOLOv8
14	2023	Deteksi Kendaraan Truk Menggunakan Tiny-YOLOv4	Tiny-YOLOv4	Deteksi objek berjalan baik	Komputasi ringan	Akurasi lebih rendah	Menggunakan YOLOv8
15	2025	Penghitung Jumlah Orang Menggunakan YOLOv8	YOLOv8	Sistem bekerja *real-time*	Akurasi tinggi	Hanya objek manusia	Menerapkan pada objek farmasi

2.2 *Image Processing*

Image Processing atau pengolahan citra merupakan proses pengolahan gambar secara digital untuk memperoleh informasi atau meningkatkan kualitas citra sehingga dapat digunakan pada tahap analisis selanjutnya. Pengolahan citra dapat mencakup berbagai proses, seperti perubahan ukuran citra (*resizing*), konversi warna, peningkatan kualitas citra, serta manipulasi citra lainnya sesuai kebutuhan sistem (Hidayah dkk., 2024).

Image Processing digunakan sebagai tahap pendukung sebelum proses deteksi objek dilakukan oleh YOLOv8n. Citra atau *frame* video yang diperoleh melalui kamera dapat disesuaikan ukuran dan formatnya agar sesuai dengan kebutuhan model. OpenCV digunakan untuk membantu proses pengambilan *frame* dari kamera, image *resizing*, konversi warna, dan pemrosesan video secara *real-time* (Wahab dkk., 2022).

2.3 *Computer Vision*

Computer Vision merupakan cabang *Artificial Intelligence* yang memungkinkan komputer mengenali dan menganalisis informasi visual dari gambar atau video. Teknologi ini dapat digunakan untuk mengenali objek dan memahami informasi yang terdapat pada suatu citra (Triyanto dkk., 2024). Perkembangan *Computer Vision* didukung oleh kemajuan kamera digital, sensor gambar, dan *Deep Learning* sehingga penerapannya berkembang pada bidang keamanan, kesehatan, pertanian, industri, dan robotika (Megantara dan Utami, 2025).

Salah satu penerapan *Computer Vision* adalah *Object Detection*, yaitu proses mendeteksi dan menentukan posisi objek pada gambar menggunakan *bounding box*. Hasil deteksi tersebut dapat digunakan sebagai dasar untuk melakukan *Object Counting* atau penghitungan jumlah objek (Putra dkk., 2024). *Computer Vision* digunakan untuk memperoleh citra objek obat melalui kamera, kemudian YOLOv8n digunakan untuk mendeteksi dan menghitung objek obat secara otomatis.

2.4 *Artificial Intelligence* (AI)

Artificial Intelligence (AI) atau kecerdasan buatan merupakan cabang ilmu komputer yang mempelajari bagaimana suatu sistem dapat dirancang agar mampu melakukan pekerjaan yang umumnya membutuhkan kecerdasan manusia. Kemampuan tersebut meliputi proses belajar *(learning*), penalaran (*reasoning*), pemecahan masalah (*problem solving*), pengambilan keputusan (*decision making*), hingga pengenalan pola (*pattern recognition*). AI dikembangkan untuk menghasilkan sistem yang dapat bekerja berdasarkan data yang diterima tanpa harus diprogram secara eksplisit pada setiap kondisi (Batubara dkk., 2024).

AI mencakup berbagai pendekatan, antara lain *Machine Learning*, *Deep Learning*, *Natural Language Processing*, *Expert System*, dan *Computer Vision*. Dalam penelitian ini, AI menjadi landasan teknologi yang digunakan untuk membangun sistem deteksi dan penghitungan objek obat.

2.5 Machine Learning

Machine Learning merupakan pendekatan dalam kecerdasan buatan yang memungkinkan sistem mempelajari pola dari data untuk menghasilkan prediksi atau keputusan. Proses pembelajaran dilakukan dengan memberikan data kepada model sehingga parameter model dapat disesuaikan berdasarkan pola yang ditemukan. *Machine Learning* dapat digunakan pada berbagai permasalahan klasifikasi, regresi, pengenalan pola, dan deteksi objek.

Dalam penelitian ini, proses pembelajaran diterapkan melalui pelatihan model YOLOv8n menggunakan dataset citra obat yang telah diberi anotasi.

Model mempelajari karakteristik visual dari kelas kapsul dan kaplet sehingga dapat digunakan untuk melakukan prediksi terhadap citra atau video baru.

2.6 *Object Detection*

Object Detection merupakan proses pada *Computer Vision* yang bertujuan untuk mengenali objek sekaligus menentukan lokasi objek dalam suatu citra. Berbeda dengan klasifikasi citra yang menghasilkan satu atau beberapa kelas untuk keseluruhan gambar, *Object Detection* memberikan informasi mengenai kelas dan posisi masing-masing objek melalui *bounding box*. Hasil tersebut dapat digunakan untuk berbagai kebutuhan, termasuk pemantauan, pengenalan objek, *tracking*, dan *Object Counting.*

Dalam penelitian ini, *Object Detection* digunakan untuk menemukan objek obat berupa kapsul dan kaplet pada citra maupun *frame* video. Setiap objek yang memenuhi nilai *confidence threshold* akan diberikan *bounding box*, label kelas, dan *Confidence Score* yang selanjutnya digunakan dalam proses penghitungan.

2.6.1 *Object Counting*

Object Counting merupakan proses menghitung jumlah objek yang terdeteksi pada suatu citra atau video. Proses penghitungan dapat dilakukan berdasarkan jumlah *bounding box* yang valid setelah proses deteksi dan penyaringan prediksi. Dalam penerapannya, *Object Counting* dapat digunakan untuk menghitung kendaraan, manusia, barang, hasil produksi, maupun objek lain yang dapat dikenali oleh model *Computer Vision*.

Pada penelitian ini, *Object Counting* dilakukan berdasarkan hasil deteksi YOLOv8n. Sistem menghitung objek berdasarkan kelas, yaitu jumlah kapsul dan

jumlah kaplet, kemudian menjumlahkannya menjadi total objek. Pendekatan ini memungkinkan proses penghitungan dilakukan secara otomatis melalui input kamera.

2.7 *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) merupakan salah satu arsitektur *Machine Learning* yang banyak digunakan dalam pengolahan citra untuk mempelajari pola dan karakteristik objek secara otomatis. CNN bekerja dengan melakukan ekstraksi fitur dari citra melalui beberapa lapisan. Secara umum, arsitektur CNN terdiri atas *input layer*, *convolutional layer*, *activation function*, *pooling layer*, dan *fully connected layer*. *Input layer* menerima citra dalam bentuk piksel, kemudian *convolutional layer* mengekstraksi fitur seperti garis, tepi, bentuk, dan tekstur menggunakan filter atau kernel. Selanjutnya, *activation function* memberikan kemampuan jaringan untuk mempelajari pola yang lebih kompleks, sedangkan *pooling layer* mengurangi ukuran fitur dengan tetap mempertahankan informasi penting. Pada bagian akhir, *fully connected layer* mengolah fitur yang telah diperoleh menjadi hasil prediksi atau klasifikasi. Dalam perkembangannya, CNN memiliki berbagai arsitektur, seperti VGG, ResNet, dan MobileNet, yang memiliki perbedaan pada struktur dan metode pengolahan fiturnya (Alzubaidi dkk., 2021).

Dalam penelitian ini, konsep CNN digunakan sebagai dasar dalam memahami proses ekstraksi fitur pada metode *Computer Vision*. Model yang digunakan secara langsung dalam penelitian adalah YOLOv8n, yaitu varian

ringan dari YOLOv8 yang digunakan untuk melakukan *Object Detection* terhadap dua kelas objek obat, yaitu kapsul dan kaplet.

2.8 *You Only Look Once* (*YOLO*)

You Only Look Once (YOLO) merupakan salah satu algoritma *Object Detection* berbasis *Deep Learning* yang dikembangkan untuk melakukan deteksi objek secara cepat. YOLO menggunakan pendekatan single-stage detector, yaitu proses deteksi lokasi dan klasifikasi objek dilakukan dalam satu proses prediksi. Pendekatan tersebut membuat YOLO sesuai untuk berbagai aplikasi *Computer Vision* yang membutuhkan pemrosesan secara *real-time* (Therino Elevan dkk., 2025).

Prinsip kerja YOLO adalah menerima citra sebagai input, kemudian memproses citra untuk memprediksi keberadaan objek. Model menghasilkan informasi berupa *bounding box*, kelas objek, dan *Confidence Score*. Informasi tersebut digunakan untuk mengetahui jenis dan lokasi objek yang terdeteksi. Hasil deteksi selanjutnya dapat digunakan untuk proses lain, seperti *Object Tracking* dan *Object Counting*.

Perkembangan algoritma YOLO terus mengalami peningkatan dari berbagai versi. Setiap perkembangan membawa perubahan pada arsitektur, akurasi, kecepatan, dan efisiensi model. Dalam penelitian ini, YOLOv8 dipilih karena sesuai dengan kebutuhan penelitian yang berfokus pada *Object Detection* dan *Object Counting* terhadap objek obat menggunakan dataset citra. YOLOv8 juga mendukung pelatihan dataset khusus serta integrasi dengan Python dan OpenCV.

2.8.1 *YOLOv8*

YOLOv8 merupakan model *Object Detection* yang dirancang untuk melakukan deteksi objek dengan cepat melalui proses inferensi. Arsitektur YOLOv8 terdiri atas beberapa komponen utama yang saling berhubungan, yaitu *backbone*, *neck*, dan *detection head*. *Backbone* berfungsi mengekstraksi karakteristik atau fitur dari citra masukan, *neck* menggabungkan fitur dari beberapa tingkat resolusi, sedangkan detection head menghasilkan prediksi berupa lokasi objek, kelas objek, dan tingkat keyakinan terhadap hasil deteksi.

Pada penelitian ini, YOLOv8 digunakan untuk mengenali dua kelas objek obat, yaitu kapsul dan kaplet. Citra yang diberikan sebagai input diproses untuk memperoleh fitur visual, kemudian fitur tersebut digunakan untuk menghasilkan prediksi objek. Hasil prediksi menjadi dasar proses deteksi dan penghitungan jumlah objek.

2.8.2 Arsitektur *YOLOv8*

Backbone bertugas mengekstraksi fitur-fitur penting dari citra masukan menggunakan jaringan CNN. Tahap ini menghasilkan representasi fitur yang digunakan pada proses deteksi objek.

Neck berfungsi menggabungkan informasi dari berbagai tingkat fitur (*multi-scale feature fusion*) sehingga model dapat mendeteksi objek dengan ukuran kecil, sedang, maupun besar secara lebih baik.

Detection head merupakan bagian akhir yang bertugas memprediksi lokasi *bounding box*, kelas objek, serta *Confidence Score*. Pada YOLOv8 digunakan pendekatan *anchor-free detection* sehingga proses prediksi menjadi lebih

sederhana dan efisien dibandingkan pendekatan berbasis anchor pada generasi sebelumnya.

2.8.3 *Bounding box*

Bounding box merupakan kotak pembatas yang digunakan untuk menunjukkan lokasi objek yang berhasil dideteksi dalam suatu citra. *Bounding box* memungkinkan sistem menentukan bagian citra yang dianggap sebagai objek oleh model dan biasanya disertai label kelas serta *Confidence Score* (Putra dkk., 2024)

Bounding box digunakan untuk menandai setiap objek kapsul dan kaplet yang berhasil dideteksi oleh YOLOv8n. Setiap *bounding box* yang valid merepresentasikan satu objek yang kemudian digunakan sebagai dasar dalam proses penghitungan.

2.8.4 *Confidence Score*

Confidence Score merupakan nilai yang menunjukkan tingkat keyakinan model terhadap hasil prediksi suatu objek. Nilai tersebut digunakan untuk menentukan seberapa yakin model bahwa suatu wilayah citra termasuk objek tertentu. Semakin tinggi nilai *confidence*, semakin besar tingkat keyakinan model terhadap prediksi yang diberikan.

Dalam sistem penelitian ini, *Confidence Score* digunakan sebagai salah satu dasar untuk menentukan apakah hasil deteksi akan ditampilkan dan dihitung. Pengaturan *confidence threshold* penting karena nilai yang terlalu rendah dapat menyebabkan prediksi yang kurang tepat, sedangkan nilai yang terlalu tinggi dapat menyebabkan objek yang sebenarnya terdeteksi justru tidak digunakan.

2.8.5 *Non-Maximum Suppression*

Non-Maximum Suppression (NMS) merupakan proses yang digunakan dalam *Object Detection* untuk mengurangi atau menghilangkan prediksi *bounding box* yang saling tumpang tindih dan mengacu pada objek yang sama. Proses ini diperlukan karena model dapat menghasilkan lebih dari satu *bounding box* untuk satu objek.

NMS bekerja dengan membandingkan tingkat *confidence* dan tingkat tumpang tindih antar*bounding box*. *Bounding box* dengan *confidence* lebih tinggi dipertahankan, sedangkan *bounding box* lain yang memiliki tumpang tindih terlalu tinggi terhadap *bounding box* tersebut dapat dihilangkan. Tingkat tumpang tindih dapat dievaluasi menggunakan *Intersection over Union* (IoU). Dalam penelitian ini, proses NMS penting karena hasil deteksi YOLOv8 digunakan sebagai dasar *Object Counting* sehingga satu objek tidak dihitung berkali-kali.

2.8.6 YOLOv8n

YOLOv8n merupakan varian nano dari keluarga YOLOv8. Varian ini memiliki ukuran model yang lebih ringan dibandingkan varian YOLOv8 lainnya sehingga sesuai untuk kebutuhan implementasi yang mempertimbangkan kecepatan proses dan penggunaan sumber daya komputasi. Pada penelitian ini, YOLOv8n digunakan sebagai model utama untuk melakukan deteksi objek kapsul dan kaplet.

Penggunaan YOLOv8n dipilih karena penelitian berfokus pada penerapan *Object Detection* dan *Object Counting* secara *real-time.* Model yang relatif ringan dapat membantu proses inference berjalan lebih cepat sehingga sesuai untuk

penggunaan melalui kamera. Pada arsip proyek, model terlatih tersedia dalam file *best.pt* dan digunakan oleh sistem untuk melakukan deteksi pada input kamera maupun gambar.

2.9 *Library* dan *Framework*

Dalam pengembangan sistem *Object Counting* secara *real-time* menggunakan YOLOv8n, penelitian memanfaatkan beberapa library dan *framework* yang berjalan dalam lingkungan pemrograman Python. Penggunaan komponen tersebut bertujuan mendukung proses pelatihan model, pengolahan citra dan video, pengolahan data numerik, penyimpanan hasil, serta pengembangan aplikasi.

2.9.1 Python

Python merupakan bahasa pemrograman tingkat tinggi yang banyak digunakan dalam pengembangan perangkat lunak, analisis data, kecerdasan buatan, *Machine Learning*, dan *Computer Vision*. Ketersediaan berbagai library membuat Python sesuai digunakan untuk membangun sistem yang mengintegrasikan YOLOv8n dengan OpenCV.

Dalam penelitian ini, Python digunakan sebagai bahasa pemrograman utama untuk menjalankan proses *training* , inference, pengolahan citra, pengelolaan data, dan pengembangan aplikasi.

2.9.2 *Ultralytics* YOLO

Ultralytics YOLO merupakan *framework* yang menyediakan implementasi model YOLO untuk berbagai kebutuhan *Computer Vision*, khususnya *Object*

Detection. *Framework* ini mendukung proses pelatihan, validasi, inference, dan evaluasi model.

Dalam penelitian ini, Ultralytics digunakan untuk mengimplementasikan dan melatih YOLOv8n menggunakan dataset objek kapsul dan kaplet.

2.9.3 OpenCV

OpenCV (*Open Source Computer Vision* Library) merupakan pustaka pemrograman sumber terbuka yang digunakan untuk mendukung pengolahan citra digital dan *Computer Vision*. OpenCV menyediakan fungsi untuk membaca gambar, mengakses kamera, memproses video, dan menampilkan hasil deteksi.

Dalam penelitian ini, OpenCV digunakan untuk memperoleh *frame* dari webcam, memproses video secara *real-time* , serta menampilkan *bounding box*, nama kelas objek, *Confidence Score*, dan hasil penghitungan.

2.9.4 NumPy

NumPy (Numerical Python) merupakan *library* Python yang digunakan untuk komputasi numerik dan pengolahan data dalam bentuk array. NumPy menyediakan struktur data array multidimensi serta berbagai operasi numerik yang efisien.

Dalam penelitian ini, NumPy digunakan sebagai *library* pendukung dalam pengolahan data citra, koordinat *bounding box*, dan informasi numerik yang dihasilkan selama proses deteksi.

2.9.5 Matplotlib

Matplotlib merupakan *library* Python yang digunakan untuk membuat visualisasi data dalam bentuk grafik maupun gambar. Dalam penelitian ini,

Matplotlib dapat digunakan untuk memvisualisasikan hasil proses pelatihan dan evaluasi model, seperti perkembangan nilai loss, precision, recall, dan metrik lainnya.

2.10 *Software* dan Perangkat Pendukung

Pengembangan sistem memerlukan perangkat lunak pendukung untuk persiapan dataset, anotasi objek, pengembangan program, dan pengujian. Perangkat yang digunakan atau disiapkan dalam penelitian meliputi Roboflow dan Visual Studio Code.

2.10.1 Roboflow

Roboflow merupakan platform berbasis web yang digunakan untuk membantu pengelolaan dataset pada bidang *Computer Vision*. Platform ini menyediakan fasilitas untuk mengunggah dataset, melakukan anotasi, melakukan data augmentation, membagi dataset, serta mengekspor dataset ke format yang sesuai dengan *framework Deep Learning*, termasuk YOLOv8.

Dalam penelitian ini, Roboflow digunakan sebagai perangkat pendukung pengelolaan dataset. Penggunaan data augmentation juga membantu meningkatkan kemampuan generalisasi model terhadap variasi pencahayaan, sudut pengambilan gambar, dan posisi objek (Iman dkk., 2025).

2.10.2 Visual Studio Code

Visual Studio Code merupakan editor kode yang digunakan untuk menulis, mengembangkan, dan menjalankan program dalam berbagai bahasa pemrograman, termasuk Python. Visual Studio Code menyediakan code editor, terminal, ekstensi pemrograman, serta dukungan *debugging*.

Dalam penelitian ini, Visual Studio Code digunakan sebagai lingkungan pengembangan untuk menulis dan menjalankan program Python yang mengintegrasikan YOLOv8n dengan OpenCV serta komponen aplikasi lainnya.

2.11 Perangkat Keras

Perangkat keras merupakan komponen fisik yang digunakan untuk mendukung proses pengembangan, pelatihan, dan pengujian sistem.

2.11.1 Kamera/*Webcam*

Kamera atau webcam merupakan perangkat keras yang digunakan sebagai sumber input berupa gambar atau video dalam sistem *Computer Vision*. Kamera memungkinkan sistem memperoleh informasi visual dari objek pada area pengamatan untuk selanjutnya diproses dan dianalisis oleh model deteksi objek.

Dalam penelitian ini, kamera atau webcam digunakan untuk mengambil gambar atau video objek obat berupa kapsul dan kaplet secara langsung. *Frame* yang diperoleh kemudian diproses menggunakan OpenCV dan diberikan sebagai input kepada model YOLOv8n.

2.11.2 Laptop/PC

Laptop atau *Personal Computer* (PC) merupakan perangkat utama untuk menjalankan proses pengembangan dan pengujian sistem. Perangkat komputasi digunakan untuk persiapan dataset, *training model*, *inference*, pengolahan citra, dan menjalankan aplikasi.

Dalam penelitian ini, laptop/PC digunakan untuk melakukan persiapan dataset, *training* YOLOv8n, pengujian model, serta menjalankan aplikasi *Object Counting* secara *real-time.*

2.12 Dataset dan Anotasi Citra

Dataset merupakan kumpulan data yang digunakan sebagai sumber informasi dalam proses pengembangan dan pelatihan model *Machine Learning* atau *Deep Learning*. Pada *Object Detection*, dataset tidak hanya terdiri dari citra, tetapi juga memerlukan informasi mengenai lokasi dan kelas setiap objek melalui anotasi.

Dalam penelitian, dataset memiliki dua kelas objek, yaitu kapsul dan kaplet. Struktur dataset menggunakan format YOLO dan file data.yaml menetapkan nc = 2 dengan kelas 0 sebagai kapsul dan kelas 1 sebagai kaplet.

2.12.1 Label/Class

Label atau class merupakan kategori yang diberikan kepada suatu objek dalam dataset untuk menunjukkan jenis objek yang terdapat pada citra. Pada proses *Object Detection*, label berfungsi sebagai informasi bagi model mengenai kelas objek yang harus dikenali (Hussain, 2024).

2.12.2 *Bounding box*

Bounding box merupakan anotasi berbentuk kotak yang digunakan untuk menunjukkan lokasi objek yang terdapat pada suatu citra. Dalam *Object Detection*, *bounding box* tidak hanya memberikan informasi mengenai jenis objek melalui label, tetapi juga menunjukkan posisi objek pada citra (Putra dkk., 2024).

Setiap objek kapsul dan kaplet dalam dataset penelitian diberikan *bounding box* sesuai dengan posisi objeknya. Dalam format anotasi YOLO, informasi *bounding box* umumnya direpresentasikan melalui class id, koordinat pusat x dan y, serta width dan height yang dinormalisasi terhadap ukuran citra.

2.13 Pembagian Dataset

Pembagian dataset merupakan proses pengelompokan data yang digunakan dalam pengembangan dan pengujian model *Machine Learning*. Pembagian dataset dilakukan agar data memiliki fungsi yang berbeda pada setiap tahapan. Secara umum, dataset dibagi menjadi data *training* , *Validation* , dan *Testing*. Data *training* digunakan untuk melatih model agar dapat mempelajari pola dan karakteristik dari data. Data *Validation* digunakan untuk memantau dan mengevaluasi performa model selama proses pengembangan. Sementara itu, data *Testing* digunakan untuk mengetahui kemampuan akhir model terhadap data yang belum digunakan dalam proses *training* maupun *Validation* .

2.13.1 *Training*

Training merupakan proses pembelajaran model menggunakan data yang telah dilengkapi dengan label atau anotasi. Pada tahap ini, model mempelajari pola, karakteristik, dan hubungan yang terdapat pada data sehingga dapat menghasilkan prediksi terhadap data baru. Semakin sesuai data *training* dengan objek yang akan dikenali, semakin baik model dalam mempelajari karakteristik objek tersebut.

2.13.2 *Validation*

Validation merupakan proses yang digunakan untuk memantau kemampuan model selama proses pengembangan. Data *Validation* tidak digunakan sebagai data utama dalam pembelajaran model, tetapi digunakan untuk mengetahui kemampuan model terhadap data yang berbeda dari data *training* .

Hasil *Validation* dapat digunakan sebagai pertimbangan dalam menentukan model dengan performa yang lebih baik.

2.13.3 *Testing*

Testing merupakan tahap pengujian yang dilakukan untuk mengetahui kemampuan akhir model dalam melakukan prediksi terhadap data yang belum digunakan pada proses *training* maupun *Validation* . Data *Testing* digunakan untuk memberikan gambaran mengenai kemampuan model dalam mengenali objek pada data baru. Oleh karena itu, data *Testing* sebaiknya berasal dari data yang terpisah dari data *training* dan *Validation* sehingga hasil pengujian dapat menggambarkan kemampuan model secara lebih objektif.