

BAB III

METODE PENELITIAN

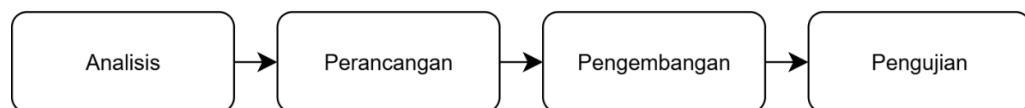
3.1 Desain Penelitian

Penelitian ini menggunakan metode Design and Development (D&D), yaitu pendekatan yang menekankan pada empat tahapan utama, yaitu analisis (*analysis*), perancangan (*design*), pengembangan (*development*), dan evaluasi (*evaluation*). Menurut Richey & Klein (2014, hal. 13), metode D&D didefinisikan sebagai *“the systematic study of design, development and evaluation processes with the aim of establishing an empirical basis for the creation of instructional and non-instructional products and tools and new or enhanced models that govern their development.”*

Berdasarkan definisi tersebut, metode D&D tidak hanya menekankan pada proses desain dan pengembangan, tetapi juga mencakup tahapan analisis serta evaluasi. Oleh karena itu, metode ini dipilih karena sesuai dengan tujuan penelitian, yaitu mengembangkan dan mengintegrasikan model deteksi YOLO ke dalam aplikasi berbasis *website* yang mampu mendeteksi, mengklasifikasikan, dan memetakan lokasi sampah plastik di sungai berdasarkan citra drone.

3.2 Prosedur Penelitian

Prosedur penelitian ini disusun berdasarkan metode *Design and Development* (D&D) sebagaimana digambarkan pada Gambar 3.1.



Gambar 3. 1 Tahapan penelitian dengan menggunakan metode D&D
(Richey & Klein, 2014)

Berdasarkan Gambar 3.1, tahap pertama adalah analisis, yang bertujuan untuk mengidentifikasi kebutuhan sistem. Setelah itu, dilakukan perancangan guna menyusun struktur sistem berdasarkan hasil analisis tersebut. Tahap berikutnya adalah pengembangan, dimana rancangan diimplementasikan menjadi sistem yang dapat dijalankan. Terakhir, dilakukan pengujian untuk

memastikan bahwa sistem yang telah dibuat bekerja sesuai spesifikasi dan memenuhi kebutuhan yang telah ditentukan.

3.3 Tahap Analisis

Tahap analisis dalam metodologi penelitian *Design and Development* (D&D) merupakan langkah awal yang bertujuan untuk memahami kebutuhan, permasalahan, serta spesifikasi sistem sebelum solusi dirancang. Pada tahap ini, peneliti menggali secara menyeluruh konteks masalah yang dihadapi, mengevaluasi kekurangan dari solusi yang telah ada, serta mengidentifikasi kebutuhan pengguna dan persyaratan teknis yang perlu dipenuhi.

Dalam penelitian ini, tahap analisis dilakukan untuk memahami tantangan utama dalam proses deteksi dan pemetaan sampah plastik yang mengambang di permukaan air sungai. Berdasarkan studi literatur dan pengamatan awal, diketahui bahwa sebagian besar penelitian sebelumnya masih memiliki keterbatasan dalam cakupan wilayah uji dan keluasan informasi yang disajikan. Beberapa studi hanya menguji sistem deteksi pada area sungai yang sempit serta menggunakan pendekatan klasifikasi sampah dalam satu kategori umum, seperti "*waste*", tanpa membedakan jenis-jenis sampah secara spesifik. Padahal, informasi jenis sampah sangat penting untuk mendukung penyusunan kebijakan pengelolaan lingkungan yang lebih tepat sasaran.

Selain itu, sebagian besar model deteksi yang dikembangkan sebelumnya tidak diintegrasikan lebih lanjut kedalam sebuah aplikasi maupun pemetaan spasial secara geografis. Hal ini menyebabkan lokasi dari objek sampah yang terdeteksi tidak diketahui dan tidak dapat diolah lebih lanjut, sehingga kurang bermanfaat informasinya. Penelitian ini berupaya untuk mengatasi keterbatasan tersebut dengan mengembangkan model deteksi yang diintegrasikan dalam bentuk *website* yang bukan hanya bisa mendeteksi sampah plastik saja, tetapi juga mengklasifikasikan kedalam lima jenis serta memetakan sebarannya berdasarkan lokasi geografis.

Untuk itu, digunakan model YOLOv11-n yang telah dilatih untuk mengenali lima jenis sampah plastik, yaitu kantong plastik, botol plastik, kemasan plastik, gelas plastik, dan styrofoam. Model ini digunakan untuk

mendeteksi objek pada citra drone yang menyimpan metadata GPS. Informasi posisi geografis latitude dan longitude dari pusat citra drone digunakan sebagai referensi untuk menghitung posisi sebenarnya dari setiap objek sampah plastik yang terdeteksi.

Konversi posisi objek dari piksel ke koordinat dunia nyata dilakukan menggunakan pendekatan *Ground Sampling Distance* (GSD). GSD adalah nilai yang merepresentasikan berapa meter di permukaan tanah yang diwakili oleh satu piksel pada citra digital. Dengan mengetahui nilai GSD, posisi deteksi dalam satuan piksel dapat dikalikan untuk menghasilkan offset dalam meter dari titik tengah citra. Offset ini kemudian digunakan untuk menghitung estimasi posisi geografis objek dengan cara menggeser koordinat GPS pusat gambar ke arah offset tersebut menggunakan library geospasial seperti Geopy.

Melalui pendekatan ini, sistem tidak hanya memberikan informasi mengenai jenis sampah plastik yang terdeteksi, tetapi juga memetakan dengan menyajikan posisi koordinat sebenarnya dalam bentuk peta yang dapat diakses juga melalui Google Maps. Visualisasi ini memungkinkan pengguna untuk memahami kondisi pencemaran sungai secara lebih menyeluruh dan akurat. Dengan demikian, hasil analisis ini menjadi dasar dalam perancangan sistem yang mampu memenuhi kebutuhan pemantauan dan pemetaan sampah plastik yang lebih efisien dan berbasis data spasial.

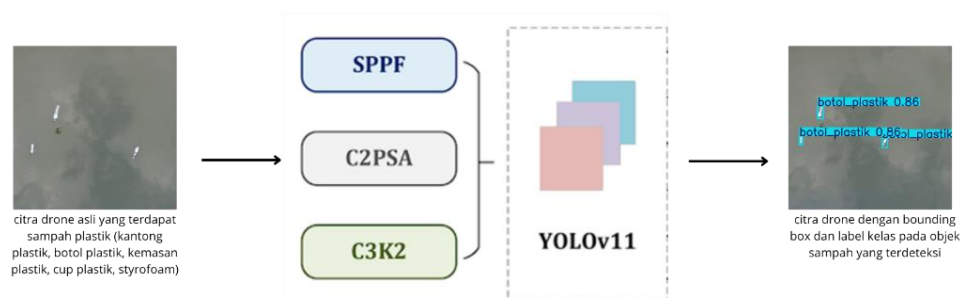
3.4 Tahap Perancangan

Tahap perancangan merupakan proses untuk menyusun rancangan teknis dari sistem yang akan dikembangkan berdasarkan hasil analisis sebelumnya. Dalam penelitian ini, tahapan perancangan difokuskan pada dua aspek utama yaitu perancangan model deteksi sampah plastik menggunakan algoritma YOLOv11-n dan perancangan aplikasi berbasis *website* sebagai media visualisasi hasil deteksi dan pemetaan distribusi sampah.

3.4.1. Perancangan Arsitektur Model

Model deteksi pada penelitian ini menggunakan algoritma YOLOv11-n (nano), yang merupakan varian ringan dari YOLOv11 dengan keunggulan dalam kecepatan dan efisiensi, sehingga cocok

diimplementasikan pada sistem berbasis web. Model ini dirancang untuk mendeteksi lima kelas sampah plastik, yaitu kantong plastik, botol plastik, kemasan plastik, gelas plastik, dan styrofoam. Model dilatih menggunakan kombinasi dataset publik dan dataset pribadi yang telah dianotasi dengan augmentasi tambahan untuk meningkatkan keberagaman dataset. Pada Gambar 3.2 ditampilkan rancangan arsitektur model serta contoh hasil deteksi pada citra drone.



Gambar 3. 2 Arsitektur Model

Pada Gambar 3.2 menunjukkan bahwa model ini memiliki tiga bagian utama, yaitu *backbone*, *neck*, dan *head*, yang bekerja sama untuk mendeteksi objek sampah plastik pada citra drone. Bagian *backbone* berfungsi mengambil ciri-ciri penting dari gambar, seperti bentuk, tepi, dan warna objek sampah plastik. Di bagian ini digunakan blok C3k2 yang mampu mengambil fokus pada bentuk kantong plastik, botol, atau kemasan meskipun ukurannya kecil atau warnanya mirip dengan latar belakang. Modul SPPF membantu model melihat objek dari berbagai ukuran sekaligus, misalnya gelas plastik kecil yang mengapung di tepi sungai dengan styrofoam yang berada di tengah sungai. Selain itu, C2PSA membuat model lebih fokus pada area yang kemungkinan terdapat sampah.

Bagian *neck* menggabungkan informasi dari berbagai ukuran gambar supaya model bisa mendeteksi sampah plastik kecil maupun besar. Modul C3k2 di bagian ini mempertahankan detail ciri-ciri objek plastik agar tidak hilang saat proses penggabungan, sedangkan C2PSA membantu mengarahkan fokus deteksi ke area dengan kemungkinan sampah lebih

tinggi. Bagian *head* adalah tahap di mana model menghasilkan prediksi akhir berupa *bounding box*, *objectness score*, dan jenis sampah plastik yang terdeteksi.

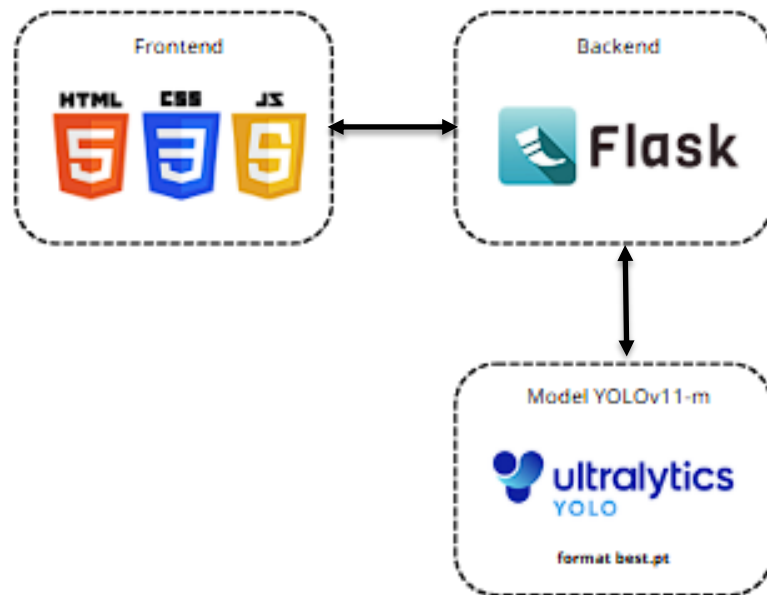
Proses deteksi dimulai dari menginputkan gambar drone, lalu melalui *backbone* untuk mengambil ciri-ciri sampah plastik, dilanjutkan ke *neck* untuk menggabungkan informasi dari berbagai skala, dan akhirnya ke *head* untuk menghasilkan prediksi. Hasil prediksi kemudian disaring dengan *Non-Maximum Suppression* (NMS) agar hanya kotak terbaik yang ditampilkan untuk setiap objek. Dengan rancangan ini, YOLOv11-n dapat mengenali berbagai jenis sampah plastik seperti kantong plastik, botol plastik, kemasan plastik, gelas plastik, dan styrofoam.

3.4.2. Perancangan Aplikasi

Perancangan aplikasi ini dirancang untuk memastikan integrasi yang optimal antara antarmuka pengguna, logika sistem, dan model deteksi objek, sehingga proses deteksi, klasifikasi, dan visualisasi data sampah plastik dapat berjalan secara cepat, akurat, dan terintegrasi dalam satu platform berbasis web.

3.4.2.1. Diagram Arsitektur Aplikasi

Diagram arsitektur aplikasi berfungsi sebagai representasi visual untuk menggambarkan struktur sistem, komponen-komponen yang terlibat, serta hubungan antar bagian. Perancangan arsitektur ini mempermudah proses pengembangan, pengambilan keputusan teknis, serta dokumentasi sistem secara keseluruhan. Gambar 3.3 menunjukkan rancangan arsitektur aplikasi berbasis *website* yang dikembangkan dalam penelitian ini.



Gambar 3. 3 Diagram arsitektur aplikasi berbasis *website*

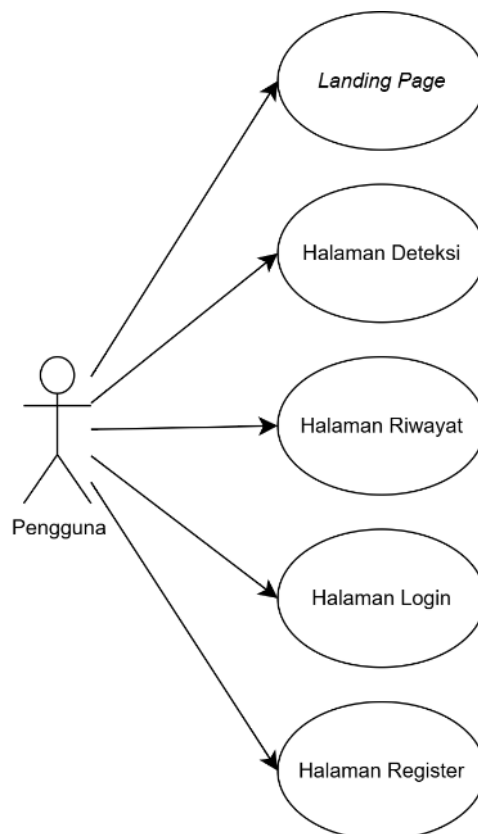
Berdasarkan Gambar 3.3, arsitektur sistem aplikasi ini terdiri dari tiga komponen, yaitu *front-end*, *back-end*, dan model deteksi objek YOLOv11-n dalam bentuk *best.pt*. Pada bagian *front-end* dibangun menggunakan HTML, CSS, JavaScript, dan *framework* Bootstrap. Pada bagian *front-end*, terdiri dari tiga halaman utama, yaitu *landing page*, halaman upload, dan halaman riwayat yang hanya dapat diakses oleh pengguna yang telah melakukan login.

Back-end menggunakan *framework* Flask Python yang berfungsi sebagai pengatur logika utama sistem. Tugas utama *back-end* meliputi menerima file gambar dari *front-end* pengguna dan menjalankan model YOLOv11-n untuk deteksi sampah plastik, serta mengirimkan kembali hasil deteksi ke *front-end* untuk ditampilkan ke pengguna.

Model deteksi objek menggunakan file hasil pelatihan YOLOv11-n berupa *best.pt*. Model ini dijalankan di sisi server melalui *back-end* Flask untuk melakukan inferensi terhadap citra yang diunggah pengguna pada *front-end*.

3.4.2.2. Use Case Diagram Aplikasi

Use case diagram adalah gambar yang menunjukkan bagaimana pengguna berinteraksi dengan sistem untuk mencapai tujuan tertentu (Pooley & King, 1999). Gambar 3.4 menampilkan *use case* diagram dari aplikasi berbasis web ini.



Gambar 3. 4 *Use case* diagram aplikasi

Penjelasan lebih rinci Gambar 3.4 mengenai masing-masing *use case* dalam aplikasi ini adalah sebagai berikut:

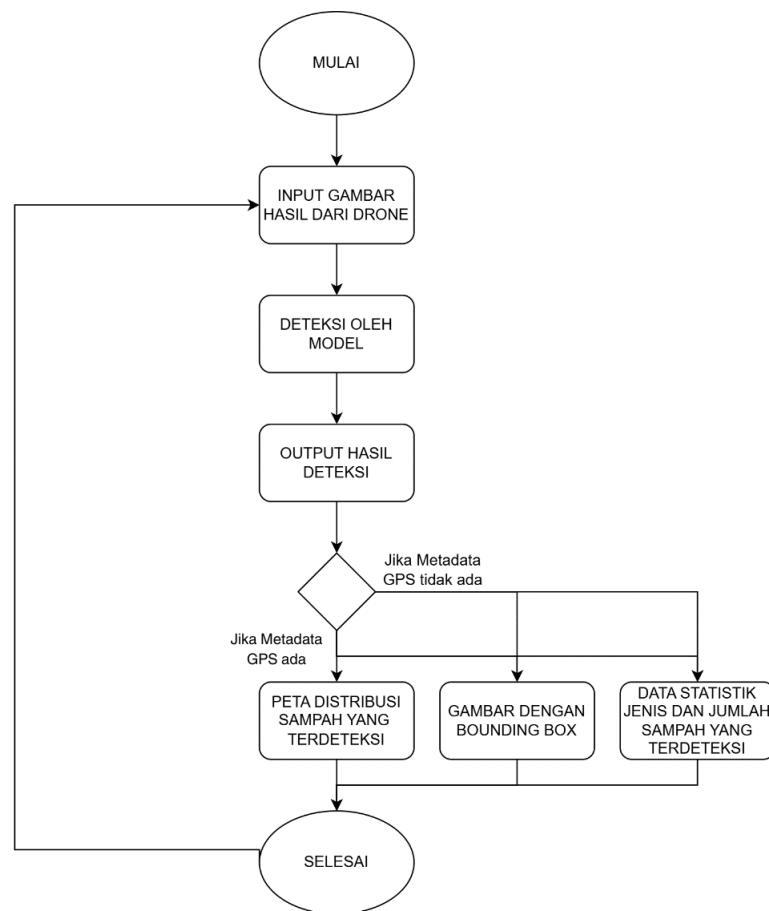
- a. *Landing page* merupakan halaman awal yang menampilkan informasi umum mengenai aplikasi, tujuan penggunaannya, serta panduan singkat tentang cara menggunakan fitur-fitur yang tersedia.
- b. Halaman deteksi memungkinkan pengguna untuk mengunggah gambar yang akan diproses oleh model deteksi. Hasil dari proses ini akan ditampilkan dalam bentuk citra drone dengan *bounding box* objek sampah

yang terdeteksi, data statistik jenis dan jumlah sampah, serta pemetaan lokasi nyata objek sampah yang terdeteksi.

- c. Halaman riwayat ini menampilkan riwayat hasil deteksi yang telah dilakukan oleh pengguna. Informasi yang ditampilkan mencakup tanggal deteksi, gambar yang diunggah, serta hasil analisis. Halaman ini hanya dapat diakses setelah pengguna *login* ke dalam sistem.
- d. Halaman *login* digunakan oleh pengguna yang sudah terdaftar untuk masuk ke dalam sistem dengan memasukkan email dan kata sandi yang telah dibuat sebelumnya.
- e. Halaman *register* merupakan halaman pendaftaran bagi pengguna baru. Di sini, pengguna dapat membuat akun dengan mengisi data seperti nama, email, dan kata sandi.

3.4.2.3. *Flowchart* Aplikasi

Flowchart aplikasi digunakan untuk menggambarkan alur proses kerja sistem secara sistematis, mulai dari input yang diberikan oleh pengguna hingga output yang ditampilkan oleh sistem. Perancangan *flowchart* ini penting untuk memahami logika kerja aplikasi secara menyeluruh dan menjadi acuan dalam proses pengembangan maupun *debugging* sistem. *Flowchart* aplikasi yang dikembangkan dalam penelitian ini ditunjukkan pada Gambar 3.5.



Gambar 3. 5 *Flowchart* aplikasi yang dikembangkan

Dapat dilihat pada Gambar 3.5, *flowchart* tersebut menjelaskan tahapan alur dari aplikasi yang dikembangkan. Pertama, pengguna mengunggah gambar hasil dari drone dari lokal komputer. Kemudian gambar diproses oleh model deteksi YOLOv11-n untuk mendeteksi dan mengklasifikasikan objek sampah plastik berdasarkan lima kelas yang telah ditentukan diantaranya kantong plastik, botol plastik, kemasan plastik, gelas plastik, dan styrofoam. Model YOLOv11-n dijalankan di sisi server *back-end* Flask yang melakukan inferensi terhadap gambar dan menghasilkan output berupa daftar objek yang terdeteksi beserta posisi *bounding box* dan kelasnya serta data statistik mengenai jumlah dan jenis sampah plastik yang terdeteksi. Jika metadata GPS tersedia dalam file gambar, sistem juga akan mengekstrak informasi

koordinat dari setiap objek terdeteksi menggunakan pendekatan perhitungan *Ground Sampling Distance* (GSD) untuk mengonversi piksel ke koordinat geografis *latitude* dan *longitude*, yang kemudian dimanfaatkan untuk memetakan lokasi sampah plastik yang terdeteksi pada peta.

Dengan demikian, sistem menghasilkan tiga keluaran utama yang akan dikirimkan kembali ke antarmuka pengguna *front-end*, yaitu:

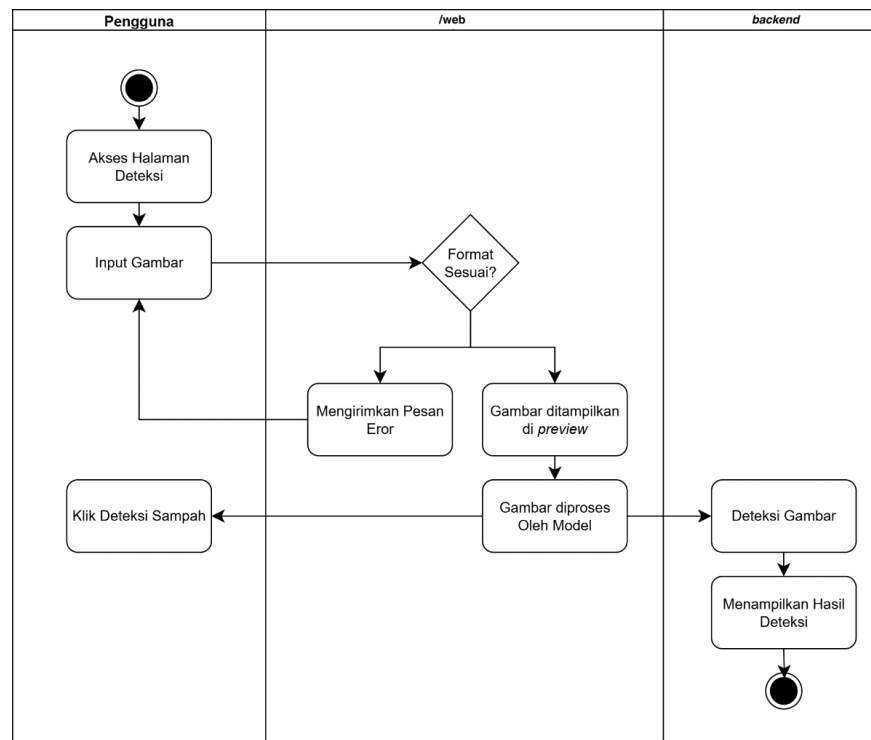
1. Citra hasil deteksi berupa yang telah diproses dan diberi *bounding box* untuk menunjukkan lokasi serta jenis sampah plastik yang terdeteksi.
2. Data statistik berupa informasi mengenai jumlah masing-masing sampah tiap kelas yang terdeteksi.
3. Peta statis berupa titik-titik lokasi geografis objek sampah plastik yang terdeteksi, jika metadata GPS tersedia pada citra.

Setelah seluruh hasil ditampilkan, pengguna dapat melanjutkan dengan mengunggah gambar lain untuk dideteksi.

3.4.2.4. Activity Diagram Aplikasi

Activity diagram merupakan salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan alur aktivitas atau proses kerja dalam suatu sistem secara sistematis (Pooley & King, 1999). Diagram ini menunjukkan urutan tindakan, proses pengambilan keputusan, serta kemungkinan percabangan alur, mulai dari kondisi awal hingga akhir dalam suatu proses atau alur kerja bisnis.

Gambar 3.6 memperlihatkan *activity* diagram utama aplikasi pada halaman deteksi, yang menggambarkan proses validasi input saat pengguna mengunggah gambar.



Gambar 3. 6 *Activity* diagram aplikasi

Pada Gambar 3.6 proses dimulai ketika pengguna mengakses halaman deteksi dan mengunggah gambar yang akan dideteksi. Sistem kemudian memeriksa format file yang diunggah. Jika format tidak sesuai dengan ketentuan, proses tidak dilanjutkan dan sistem akan menampilkan pesan kesalahan kepada pengguna. Namun, jika format file valid, sistem akan menampilkan *preview* dari gambar yang diunggah. Selanjutnya, ketika pengguna menekan tombol "Deteksi Sampah", sistem akan memproses gambar menggunakan model deteksi YOLOv11-n dan mengembalikan hasilnya kepada pengguna. Hasil yang ditampilkan mencakup gambar dengan *bounding box* yang menunjukkan jenis dan posisi sampah plastik yang terdeteksi, data statistik mengenai jumlah sampah dari tiap kelas yang terdeteksi, serta peta lokasi objek yang terdeteksi. Setelah hasil ditampilkan, pengguna dapat memilih untuk mengunggah gambar lain untuk dideteksi kembali.

3.5 Tahap Pengembangan

Tahap pengembangan merupakan proses implementasi dari rancangan sistem yang telah disusun pada tahap sebelumnya. Dalam penelitian ini, pengembangan difokuskan pada dua aspek utama, yaitu pengembangan model deteksi sampah plastik menggunakan YOLOv11-n dan pengembangan aplikasi berbasis *website* untuk menampilkan hasil deteksi secara visual. Seluruh proses dilakukan di lingkungan lokal menggunakan perangkat keras dan perangkat lunak pendukung yang tersedia.

Perangkat keras yang digunakan terdiri dari komputer untuk pelatihan model serta drone. Spesifikasi perangkat keras ditunjukkan pada Tabel 3.1.

Tabel 3. 1 Perangkat Keras

No	Nama	Spesifikasi
1	Komputer	Prosesor Intel Core i9-11900 (11th Gen), GPU NVIDIA GeForce RTX 3080 Ti, RAM 30 GB
2	Drone	DJI Mini SE, Kamera 12 MP, <i>Image Sensor</i> 1/2.3Inch CMOS.

Berdasarkan Tabel 3.1, komputer digunakan untuk proses pelatihan model YOLOv11-n menggunakan dataset citra sampah yang telah dikumpulkan. Sementara itu, drone digunakan dalam proses pengambilan dataset pribadi. Selain sebagai perangkat pengumpulan data pribadi, drone juga dimanfaatkan dalam tahap pengujian model dengan menghasilkan citra uji sebagai input untuk mendeteksi keberadaan sampah plastik yang mengapung di permukaan sungai.

Selain perangkat keras, proses pengembangan sistem ini juga melibatkan berbagai perangkat lunak dengan versi tertentu yang mendukung pelatihan model, pelabelan data, serta pengembangan aplikasi web. Rincian perangkat lunak yang digunakan dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Perangkat Lunak

No	Nama	Versi
1	Python	3.11.9
2	YOLO	11-n
3	LabelImg	1.8.1
4	Visual Studio Code	1.102.3
5	torch	2.5.1
6	cuDNN	9.11.0

Berdasarkan Tabel 3.2 diatas, Python digunakan sebagai bahasa pemrograman utama, YOLO 11-n sebagai framework pelatihan dan inferensi, LabelImg untuk anotasi data, Visual Studio Code sebagai IDE, sedangkan Torch dan cuDNN mendukung komputasi GPU. Kombinasi perangkat lunak tersebut memungkinkan sistem dikembangkan secara terintegrasi dari tahap pelabelan hingga implementasi aplikasi *website*.

3.5.1. Pengembangan Model

Pengembangan model YOLOv11-n dalam penelitian ini dilakukan melalui beberapa tahap diantaranya persiapan dataset, augmentasi dataset, pembagian dataset, dan pelatihan model. Setiap tahap dirancang untuk memastikan model mampu mendeteksi berbagai jenis sampah plastik dengan akurasi yang baik.

3.5.1.1. Persiapan Dataset

Dataset yang digunakan dalam pelatihan model berasal dari dua sumber utama, yaitu:

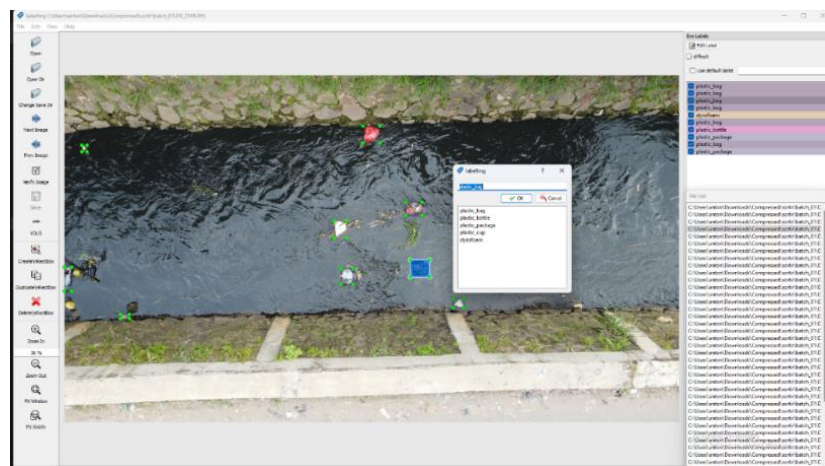
a. Dataset publik

Sebanyak 942 gambar diperoleh dari sumber dataset terbuka yang relevan, yang memuat objek-objek sampah plastik yang mengapung di permukaan sungai yang diambil dari atas tegak lurus ke bawah (*nadir view*). Dataset ini dipilih karena kesesuaian objek dengan tujuan penelitian (Louise, S., dkk, 2022; Wenlong Han, 2021; Yang Zhe, 2023).

b. Dataset pribadi

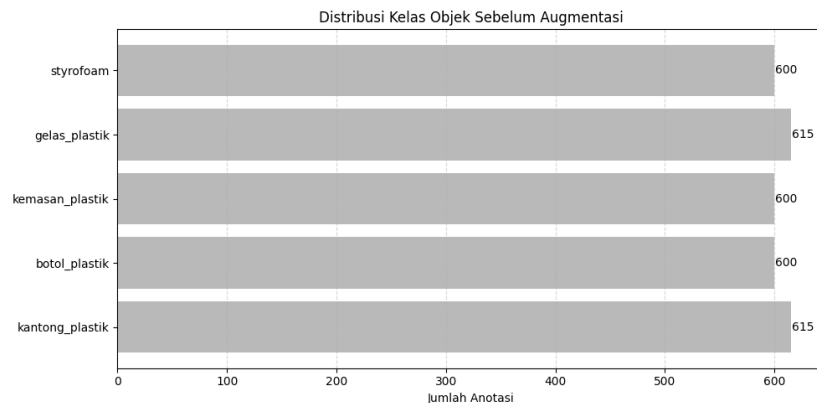
Sebanyak 300 gambar dikumpulkan pribadi. Pengambilan gambar dilakukan sama dengan dataset publik dengan posisi kamera menghadap tegak lurus ke bawah (*nadir view*), sehingga menghasilkan citra dengan sudut pandang vertikal.

Seluruh gambar dari kedua sumber dataset kemudian melalui proses pelabelan secara manual menggunakan perangkat lunak LabelImg. Proses pelabelan dilakukan dengan menandai objek sampah plastik pada gambar menggunakan *bounding box* dan mengklasifikasikannya ke dalam lima kelas, yaitu kantong plastik, botol plastik, kemasan plastik, gelas plastik, dan styrofoam. File anotasi disimpan dalam format YOLO dengan ekstensi .txt untuk mendukung kompatibilitas langsung saat pelatihan model. Contoh proses anotasi dataset menggunakan *LabelImg* ditunjukkan pada Gambar 3.7.



Gambar 3. 7 Proses pelabelan pada *software LabelImg*

Setelah tahap anotasi dataset selesai, didapatkan jumlah distribusi dataset sebelum dilakukan proses augmentasi ditampilkan pada Gambar 3.8.



Gambar 3. 8 Distribusi dataset sebelum augmentasi

Gambar 3.8 menunjukkan distribusi anotasi pada dataset sebelum dilakukan augmentasi, yaitu sebanyak 615 anotasi untuk kelas kantong plastik, 600 untuk botol plastik, 600 untuk kemasan plastik, 615 untuk gelas plastik, dan 600 untuk styrofoam dalam 1.242 gambar.

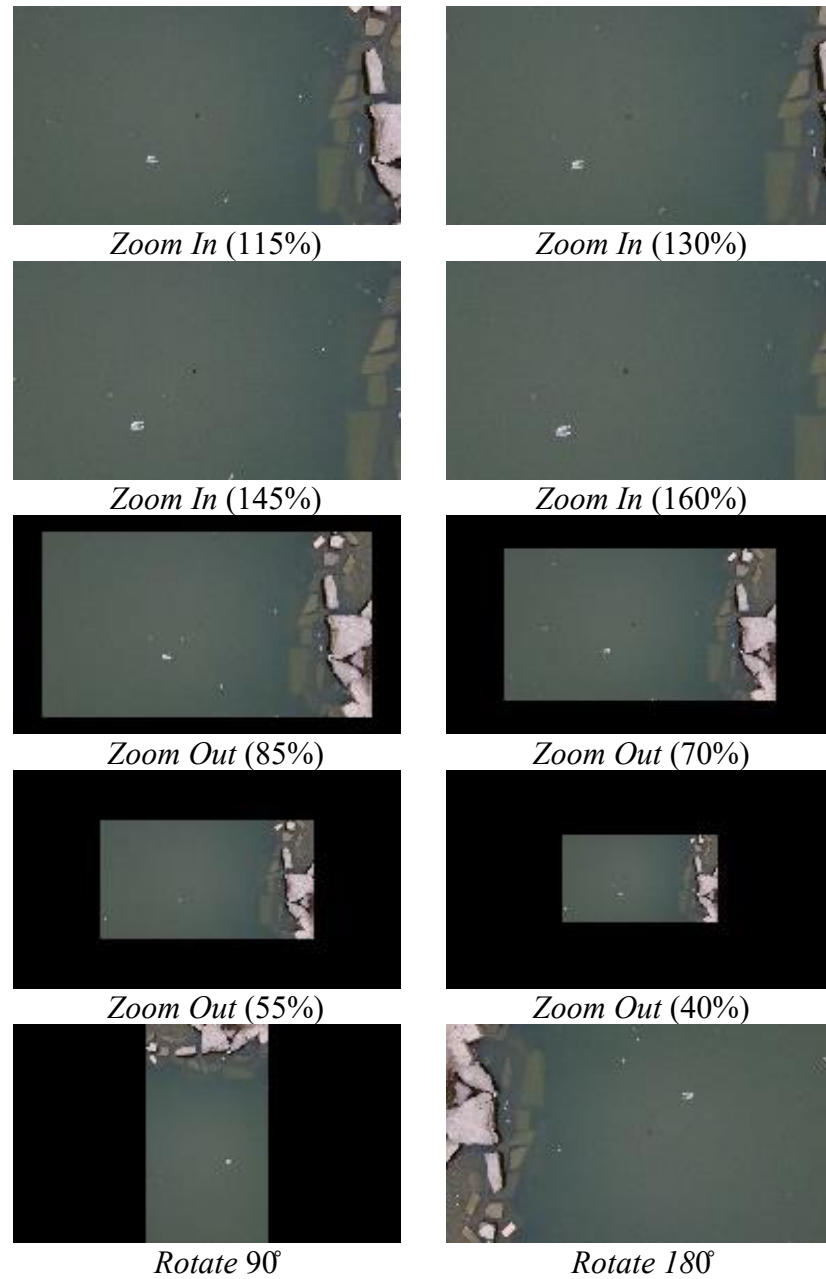
3.5.1.2. Augmentasi Dataset

Untuk meningkatkan kemampuan generalisasi model terhadap objek dengan ukuran dan orientasi yang bervariasi, dilakukan proses augmentasi data secara otomatis menggunakan skrip Python. Augmentasi dilakukan setelah pengumpulan dataset sebanyak 1.242 gambar gabungan dataset publik dan dataset pribadi. Teknik augmentasi yang digunakan meliputi:

- Zoom in* dengan skala 115%, 130%, 145%, dan 160%
- Zoom out* dengan skala 85%, 70%, 55%, dan 40%
- Rotasi dengan sudut 90° dan 180°

Augmentasi dilakukan dengan tujuan mensimulasikan kondisi pengambilan gambar dari berbagai sudut dan jarak ketinggian mengingat pengujian model akan diuji berdasarkan ketinggian yang bervariasi dari 5 meter – 10 meter. Proses ini juga berperan penting dalam memperkaya variasi data, sehingga model dapat mengenali objek sampah plastik baik yang tampak

kecil dari kejauhan maupun besar dari jarak dekat. Teknik augmentasi yang digunakan dapat dilihat pada Gambar 3.9.

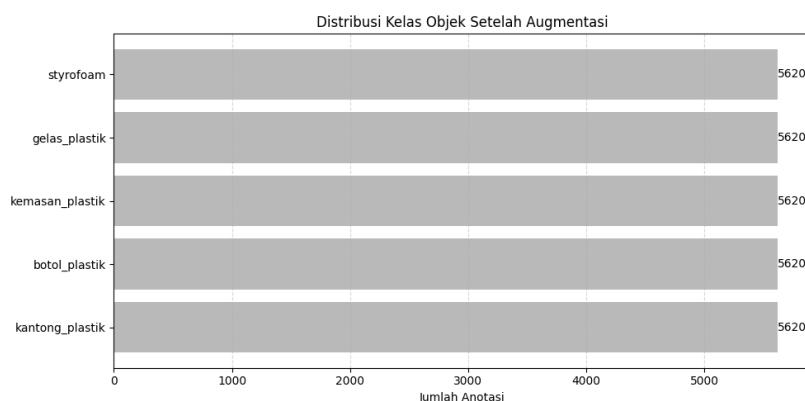


Gambar 3. 9 Teknik augmentasi

Gambar 3.9 memperlihatkan berbagai teknik augmentasi yang diterapkan untuk meningkatkan variasi data dalam pelatihan model. Teknik augmentasi dilakukan pada setiap citra asli dengan teknik *zoom in*, *zoom out*, serta *rotate*. Tujuannya adalah untuk mensimulasikan kondisi nyata pengambilan

gambar dari drone, yang memiliki ketinggian berbeda-beda. Misalnya, *zoom in* diterapkan dalam skala 115%, 130%, 145%, dan 160%. Sedangkan *zoom out* diterapkan dalam skala 85%, 70%, 55%, dan 40% untuk meniru perbedaan jarak ketinggian antara drone dan objek. Adapun rotasi sebesar 90° dan 180° digunakan untuk merepresentasikan kemungkinan perubahan orientasi kamera.

Hasil dari proses augmentasi tersebut berhasil memperluas jumlah dataset dari 1.242 gambar awal menjadi 11.651 gambar. Setiap kelas objek disesuaikan agar memiliki jumlah anotasi yang seimbang, sehingga tidak ada dominasi kelas tertentu yang dapat menimbulkan bias dalam proses pelatihan. Distribusi akhir anotasi per kelas dapat dilihat pada Gambar 3.10.



Gambar 3. 10 Distribusi dataset setelah diaugmentasi

Gambar 3.10 memperlihatkan bahwa jumlah anotasi untuk setiap kelas objek telah diseimbangkan, masing-masing mencapai 5.620 anotasi. Dengan jumlah anotasi yang seragam di seluruh kelas, model diharapkan dapat melakukan deteksi dengan akurasi yang lebih merata terhadap semua jenis sampah plastik yang terdapat di dalam citra.

3.5.1.3. Pembagian Dataset

Setelah proses anotasi dan augmentasi selesai, seluruh dataset kemudian dibagi menjadi tiga subset utama dengan rasio:

- a. *Training set* : 80%
- b. *Validation set* : 10%
- c. *Test set* : 10%

Pembagian ini mengacu pada praktik standar dalam pelatihan model *deep learning*, dengan tujuan untuk mengoptimalkan proses pelatihan sekaligus memastikan evaluasi model dilakukan secara objektif dan menyeluruh. Subset *training* digunakan untuk melatih model, subset *validation* digunakan untuk memantau performa model selama pelatihan dan mencegah *overfitting*, sedangkan subset *test* digunakan untuk mengukur kemampuan generalisasi model terhadap data baru yang belum pernah dilihat sebelumnya. Distribusi kelas pada masing-masing subset setelah proses *splitting data* dapat dilihat pada Tabel 3.3.

Tabel 3. 3 Distribusi kelas tiap subset

Dataset	<i>Train</i>	<i>Val</i>	<i>Test</i>
Jumlah Gambar	9747	1218	1219
Anotasi Kelas 0	4035	476	489
Anotasi Kelas 1	4026	441	533
Anotasi Kelas 2	4005	531	464
Anotasi Kelas 3	4015	503	481
Anotasi Kelas 4	3926	537	537

Berdasarkan Tabel 3.3, data pada setiap subset memiliki distribusi yang cukup seimbang untuk tiap kelas. Keseimbangan ini penting agar model dapat belajar dengan baik saat dilatih, divalidasi, maupun diuji, sehingga hasil evaluasi lebih akurat dan model mampu bekerja dengan baik pada data baru.

3.5.1.4. Pelatihan Model

Pelatihan model dilakukan menggunakan algoritma YOLOv11-n, salah satu varian YOLOv11 yang cocok untuk penelitian yang memerlukan kecepatan deteksi tinggi dan penggunaan sumber daya ringan, sehingga ideal untuk integrasi ke dalam *website*.

Pelatihan dilakukan menggunakan skript Python `training.py` yang dijalankan melalui lingkungan pengembangan Visual Studio Code. Parameter-parameter pelatihan yang digunakan dapat dilihat pada Tabel 3.4

Tabel 3. 4 Parameter *training*

Parameter	Nilai
<i>epochs</i>	100
<i>imgsz</i>	1024
<i>batch</i>	16
<i>workers</i>	4
<i>device</i>	0
<i>patience</i>	10
<i>save</i>	<i>True</i>
<i>optimizer</i>	SGD
<i>amp</i>	<i>True</i>
<i>pretrained</i>	<i>True</i>
<i>Learning rate</i>	0.001
<i>weight_decay</i>	0.0005
<i>label_smoothing</i>	0.1
<i>freeze</i>	0
<i>mosaic</i>	<i>False</i>
<i>augment</i>	<i>False</i>
<i>rect</i>	<i>True</i>

Tabel 3.4 menampilkan konfigurasi parameter pelatihan model YOLOv11-n, yang mencakup jumlah *epoch* sebanyak 100, ukuran gambar sebesar 1024×1024 piksel, dan ukuran *batch* sebesar 16. Model dilatih menggunakan *optimizer Stochastic Gradient Descent* (SGD) dengan *learning rate* sebesar 0.001 dan *weight decay* sebesar 0.0005. Untuk meningkatkan generalisasi, digunakan label *smoothing* sebesar 0.1. Pelatihan dilakukan tanpa menerapkan teknik augmentasi tambahan, seperti *mosaic* dan *augment*, yang keduanya diset ke *False*. Parameter *pretrained=True* menunjukkan bahwa pelatihan menggunakan bobot awal dari model pralatih, sedangkan *freeze=0* berarti tidak ada lapisan yang dibekukan selama pelatihan. Sementara itu, *rect=True* menandakan bahwa rasio aspek asli gambar pada dataset tetap dipertahankan selama pelatihan, sehingga menghindari perubahan bentuk atau proporsi akibat pemaksaan ke bentuk persegi.

Setelah proses pelatihan selesai, keluaran menghasilkan file model terbaik dengan nama *best.pt*. File ini selanjutnya digunakan dalam pengembangan aplikasi web untuk melakukan inferensi atau prediksi terhadap gambar yang diunggah oleh pengguna.

3.5.2. Pengembangan Aplikasi

Pengembangan aplikasi dilakukan untuk menampilkan hasil deteksi secara visual dan interaktif melalui antarmuka berbasis web. Aplikasi ini dirancang agar pengguna dapat dengan mudah mengunggah gambar, melihat hasil deteksi objek sampah plastik, serta memahami distribusi lokasi deteksi melalui peta. Aplikasi dibangun menggunakan teknologi sebagai berikut:

1. *Back-end*: Flask Python, digunakan untuk menangani logika server, proses unggah gambar, serta menjalankan inferensi menggunakan model deteksi.

2. *Front-end*: HTML, CSS, JavaScript, dan Bootstrap digunakan untuk membangun antarmuka pengguna yang sederhana dan responsif.
3. Visualisasi Peta: Folium dan Leaflet, digunakan untuk menampilkan lokasi objek yang terdeteksi pada peta jika metadata GPS tersedia dalam citra.
4. Integrasi Model: Model YOLOv11-n dijalankan pada sisi *back-end* setiap kali pengguna mengunggah gambar untuk mendeteksi objek sampah.

Aplikasi menampilkan tiga keluaran utama kepada pengguna diantaranya:

1. Citra hasil deteksi menampilkan gambar yang telah diproses, dilengkapi dengan *bounding box* dan label jenis sampah.
2. Data statistik deteksi menampilkan jumlah objek terdeteksi per kelas dalam bentuk teks, misalnya: botol plastik: 8 objek
3. Peta berisi titik lokasi sampah yang terdeteksi, jika metadata GPS tersedia.

Proses pengembangan aplikasi dilakukan secara bertahap dengan merujuk pada *flowchart* dan arsitektur sistem yang telah dirancang sebelumnya. Pendekatan ini digunakan untuk memastikan bahwa setiap fitur berfungsi sesuai tujuan sistem dan mendukung kebutuhan visualisasi hasil deteksi secara menyeluruh dan informatif.

3.6 Tahap Pengujian

Tahap pengujian merupakan proses evaluasi terhadap sistem yang telah dikembangkan untuk memastikan seluruh komponen berjalan sesuai dengan tujuan dan kebutuhan penelitian. Pengujian difokuskan pada dua aspek utama, yaitu model deteksi objek berbasis YOLOv11-n dan aplikasi *website* yang digunakan untuk menampilkan hasil deteksi serta pemetaan lokasi objek sampah yang terdeteksi.

Pengujian dilakukan menggunakan citra lapangan yang diambil secara langsung dengan drone DJI Mini SE pada sungai dengan variasi ketinggian 5

hingga 10 meter. Tujuan dari pengujian ini adalah untuk mengevaluasi ketinggian optimal bagi model dalam melakukan deteksi pada kondisi nyata, serta menilai efektivitas aplikasi dalam menampilkan hasil deteksi.

3.6.1. Pengujian Model

Pengujian model YOLOv11-n bertujuan untuk menilai performa deteksi secara objektif sebelum model diintegrasikan ke dalam sistem aplikasi *website*. Proses ini dilakukan secara terpisah menggunakan skrip `test.py`, yang dirancang untuk menguji performa deteksi model terhadap gambar uji secara langsung, tanpa melibatkan komponen aplikasi web. Model yang digunakan merupakan model hasil pelatihan terbaik yang disimpan dalam berkas `best.pt`

Dataset pengujian terdiri dari citra yang tidak pernah digunakan selama proses pelatihan. Dataset ini mencakup lima kelas sampah plastik yaitu kantong plastik, botol plastik, kemasan plastik, gelas plastik, dan styrofoam. Proses pengujian dilakukan melalui dua pendekatan evaluasi, yaitu:

1. Evaluasi kualitatif

Evaluasi kualitatif dilakukan dengan mengamati visualisasi hasil deteksi terhadap gambar uji. Gambar-gambar uji diproses menggunakan skrip `test.py`, dan hasilnya disimpan dalam bentuk citra baru yang telah dilengkapi *bounding box* dan label kelas hasil prediksi. Aspek yang diamati dalam evaluasi kualitatif antara lain:

- a. Ketepatan posisi *bounding box*, apakah kotak deteksi mengelilingi objek dengan baik,
- b. Kecocokan label kelas, apakah prediksi kelas sudah sesuai dengan objek yang sebenarnya,

Tabel 3.5 merupakan skrip `test.py` yang digunakan dalam proses pengujian model.

Tabel 3. 5 Skrip test.py

Kode	Keterangan
<pre> from ultralytics import YOLO import os from pathlib import Path import cv2 def main(): # Path ke model model_path= Path('runs/train/yolov11n_beres/weights/best.pt') if not model_path.exists(): print(f'Model tidak ditemukan: {model_path}') return # Load model model = YOLO(model_path) model.info() # Gambar untuk prediksi image_input = 'test_images/' output_dir 'runs/predict/yolov11n_semua_kelas' results = model.predict(source=image_input, imgsz=1024, conf=0.25, iou=0.45, save=True, save_txt=False, project='runs/predict', name='yolov11n_semua_kelas', device='cpu', verbose=False) print(f' Deteksi selesai. Hasil tersimpan di: {output_dir}') if __name__ == '__main__': main() </pre>	Skrip pengujian model sebelum model diimplementasikan kedalam sebuah <i>website</i>

Pada Tabel 3.5, skrip akan mendeteksi semua gambar di folder `dataset_uji/images` menggunakan model `best.pt`, kemudian menyimpan hasil visualisasi deteksi ke dalam folder `hasil_deteksi`. Visualisasi ini menjadi bahan evaluasi kualitatif sebelum model digunakan dalam aplikasi berbasis web.

2. Evaluasi Kuantitatif

Evaluasi kuantitatif bertujuan untuk mengukur kinerja dan performa model secara numerik. Beberapa metrik pengujian yang digunakan diantaranya:

a. *Precision*

Precision menunjukkan ketepatan prediksi positif yang dihasilkan oleh model. Berikut persamaan dalam menghitung *precision*:

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

b. *Recall*

Recall menunjukkan sejauh mana model dapat mendeteksi semua objek dari satu kelas tertentu yang sebenarnya ada. Berikut persamaan dalam menghitung *recall*:

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

c. *F1-score*

F1-score merupakan rata-rata harmonik dari *precision* dan *recall*. Metrik ini sangat berguna saat menghadapi dataset yang tidak seimbang. Berikut persamaan dalam menghitung *F1-score*:

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

d. *Mean Average Precision* (mAP)

mAP merupakan rata-rata dari *average precision* pada setiap kelas. Nilai mAP digunakan untuk mengetahui kinerja model secara keseluruhan dalam mendeteksi

objek dari berbagai kelas. Berikut persamaan dalam menghitung mAP:

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i \quad (5)$$

3.6.2. Pengujian Aplikasi

Pengujian aplikasi bertujuan untuk memastikan bahwa sistem deteksi dan pemetaan sampah plastik berbasis web telah berjalan sesuai dengan perancangan. Pengujian dilakukan dengan metode *black-box testing*, yaitu menguji keluaran sistem berdasarkan masukan yang diberikan tanpa mengetahui struktur kode internal. Fitur yang diuji meliputi:

1. Halaman *Register* memastikan pengguna dapat mendaftarkan akun baru dan data disimpan ke sistem.
2. Halaman *Login* memastikan proses autentikasi berjalan sesuai dan pengguna dapat masuk menggunakan akun terdaftar.
3. Halaman Deteksi menguji proses unggah gambar, inferensi model, dan tampilan hasil deteksi.
4. Statistik Deteksi menampilkan jumlah objek terdeteksi per kelas dalam bentuk teks.
5. Visualisasi Peta menampilkan lokasi objek yang terdeteksi pada peta apabila metadata GPS tersedia dalam citra.
6. Halaman Riwayat memastikan pengguna yang login dapat melihat riwayat deteksi yang pernah dilakukan.

Tabel 3.6 merupakan instrumen pengujian aplikasi yang digunakan untuk mengevaluasi sistem berdasarkan beberapa parameter, seperti fitur yang diuji, skenario pengujian, output yang diharapkan, dan output yang dihasilkan. Setiap hasil pengujian baik, berhasil maupun tidak dicatat secara sistematis. Berdasarkan data hasil pengujian tersebut, dilakukan evaluasi untuk menilai apakah aplikasi pendeteksi dan pemetaan sampah plastik yang dibangun telah berfungsi sesuai dengan perancangan dan tujuan penelitian.

Tabel 3. 6 Instrumen pengujian aplikasi menggunakan black-box testing

No	Uji Kasus	Skenario	Hasil Yang Diharapkan
1	Register Akun	Mengisi form register dengan data yang valid dan submit	Akun berhasil dibuat dan disimpan ke sistem
2	Login Pengguna	Memasukan <i>username</i> dan <i>password</i> yang valid	Pengguna berhasil login dan diarahkan ke <i>landing page</i> atau beranda
3	Pindah ke halaman deteksi	Klik navbar Deteksi atau tombol mulai deteksi	Beralih ke halaman deteksi
4	Cara kerja <i>website</i>	Klik tombol lihat demo	Beralih ke <i>section</i> gambar contoh hasil deteksi
5	Atur ketinggian atau <i>altitude</i>	Klik panah atas atau bawah	Nilai <i>altitude</i> berubah
6	Upload Gambar	Klik tombol pilih file atau <i>drag and drop</i> gambar di area	Gambar berhasil diunggah dan <i>preview</i> gambar ditampilkan di antarmuka
7	Hapus Gambar Upload	Klik tombol Hapus	Gambar terhapus dari <i>preview</i>

No	Uji Kasus	Skenario	Hasil Yang Diharapkan
8	Proses Deteksi	Klik tombol deteksi	Muncul informasi deteksi berupa data statistik mengenai jumlah dan jenis sampah yang terdeteksi, gambar dengan <i>bounding box</i> sampah yang terdeteksi, serta peta lokasi dari tiap titik sampah yang terdeteksi.
9	<i>Error handling</i> input gambar	Input gambar bukan .jpg atau .png lalu klik deteksi	Muncul pesan Format file tidak didukung.
10	Lokasi <i>direct</i> ke <i>Google Maps</i>	Klik titik yang berada di peta, lalu klik buka di maps	Langsung <i>direct</i> ke lokasi di <i>google maps</i>
11	Halaman Riwayat Deteksi	Pengguna sudah login sebelumnya, dan klik icon profile dan klik riwayat.	Beralih ke halaman riwayat deteksi. Halaman ini menyimpan hasil deteksi pengguna sebelumnya.
12	Log out	Pengguna yang telah login sebelumnya, klik icon profil dan klik keluar	Berhasil keluar dan terdapat notifikasi Berhasil logout, kemudian beralih ke halaman beranda atau <i>landing page</i> .

Selain pengujian fungsionalitas aplikasi, dilakukan juga uji lapangan untuk menilai performa sistem dalam kondisi yang menyerupai situasi nyata. Uji ini melibatkan pengambilan citra menggunakan drone di area

sungai dengan variasi jenis kelas sampah dan pada beberapa ketinggian pengambilan citra pada dua titik lokasi berbeda. Citra yang diperoleh kemudian diproses melalui aplikasi *website* untuk mendeteksi objek secara otomatis. Untuk mengevaluasi hasil, dilakukan perhitungan manual jumlah sampah pada citra drone sebagai acuan pembandingan. Tabel 3.7 berikut menyajikan instrumen uji lapangan berdasarkan kelas objek dan ketinggian pengambilan citra.

Tabel 3. 7 Uji Model Berdasarkan Kelas dan Ketinggian

No	Kelas	Ketinggian (m)	Hasil Deteksi Model	Hasil Pengamatan Langsung Gambar
1	Kantong Plastik	5		
		6		
		7		
		8		
		9		
		10		
2	Botol Plastik	5		
		6		
		7		
		8		
		9		
		10		
3	Kemasan Plastik	5		
		6		
		7		
		8		
		9		
		10		
4		5		

No	Kelas	Ketinggian (m)	Hasil Deteksi Model	Hasil Pengamatan Langsung Gambar
	Cup Plastik	6		
		7		
		8		
		9		
		10		
5	Styrofoam	5		
		6		
		7		
		8		
		9		
		10		
6	Semua Kelas	5		
		6		
		7		
		8		
		9		
		10		

Pengujian yang dirangkum dalam Tabel 3.7 bertujuan untuk mengevaluasi kemampuan model dalam mengenali berbagai jenis sampah plastik pada kondisi ketinggian yang berbeda. Hasil deteksi kemudian dibandingkan dengan perhitungan manual objek sampah yang terdapat dalam citra drone untuk melihat kesesuaian model dalam mendeteksi sampah plastik, sehingga didapatkan gambaran mengenai performa model dan ketinggian optimal untuk digunakan dalam mendeteksi dan pemetaan sampah plastik di sungai.