Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.1, February 2026, pp. 284~297
e-ISSN: 2548-6861

Real-Time Waste Detection System Using YOLOv12 with Transfer Learning

Adellia Jovina [1]*, Ester Lumba [2]*
* Department of Informatics, University of Bunda Mulia, Indonesia
s32220025@student.ubm.ac.id [1], l0178@lecturer.ubm.ac.id [2]

Article Info

Article history:

Received 2025-12-17
Revised 2026-01-08
Accepted 2026-01-13

Keyword:

YOLOv12,
Waste Detection,
Transfer Learning,
Computer Vision,
Web-Based System.

ABSTRACT

Waste sorting at the source remains a major challenge in Indonesia due to limited public awareness and the absence of accessible tools for waste classification. While YOLO-based object detection has been widely applied for waste detection, the adoption of the latest YOLO architecture in web-based, real-time public-oriented systems remains limited. This study aims to develop and experimentally evaluate a web-based waste detection system using YOLOv12 with a transfer learning approach to classify waste into organic, inorganic, and hazardous (B3) categories along with their subcategories. The system was developed using the Flask framework and supports image upload and real-time camera-based detection. A real-world dataset was annotated and divided into training, validation, and testing sets for experimental evaluation. The proposed model achieved a precision of 0.86, recall of 0.74, mAP@0.5 of 0.83, and mAP@0.5:0.95 of 0.68, with an average inference time of 0.0187 seconds per image (53.40 FPS). Overall, these results indicate that YOLOv12 with transfer learning provides an effective balance between accuracy and inference speed for web-based real-time waste detection systems, supporting its applicability for practical waste sorting solutions.

I. Introduction

Waste management in Indonesia remains a major challenge in environmental management. Based on data from the National Waste Management Information System (SIPSN), the amount of waste generated in Indonesia in 2024 exceeded 34 million tons, with approximately 40% of the total not being properly managed [1]. This condition leads to various environmental impacts and increases the workload of officers at Final Disposal Sites (TPA), particularly due to the time-consuming and inefficient waste sorting process.

On the other hand, public understanding of waste categories such as organic, inorganic, and hazardous (B3) waste is still limited. Many individuals have difficulty identifying the correct classification, causing waste that has already been sorted to be mixed again. In addition, interviews with TPA officers revealed that the public often feels confused about what actions should be taken after waste has been sorted. These issues indicate the need for a supporting tool that can help the community accurately identify waste categories and determine appropriate follow-up actions for the resulting sorted waste.

Advancements in computer vision, particularly the YOLO (You Only Look Once) algorithm, enable the development of image-based systems capable of recognizing objects and providing information about waste categories. YOLO is an end-to-end convolutional neural network-based object detection framework that detects objects across the entire image in a single processing stage, eliminating the need for region proposal mechanisms [2]. This approach enables high inference speed while maintaining competitive detection accuracy, making YOLO well suited for real-time systems.

Several previous studies have applied YOLOv3, YOLOv5, YOLOv8, and YOLOv11 for waste detection. Soerya *et al.* applied YOLOv3 to identify and classify office waste [3]. The system was developed on an NVIDIA Jetson Nano device to support real-time processing. The results showed that YOLOv3 was able to detect waste objects with good accuracy even when running on embedded devices. These findings demonstrate that YOLO can be efficiently implemented in systems with limited computational resources.

Research conducted by Abdillah *et al*. developed a web-based system for classifying organic and inorganic waste using YOLOv5 [4]. The system allows users to perform waste detection through a web interface by directly uploading images. The results showed that the integration of YOLOv5 into a web-based system was able to provide accurate and responsive classification results. This study indicates that YOLO can be utilized to support practical and easily accessible waste sorting for users.

Suwela and Hedriyadi applied a transfer learning approach using YOLOv8 to detect types of waste [5]. The model was trained by leveraging pretrained weights, enabling improved detection performance compared to training from scratch. Evaluation results showed that the use of transfer learning increased accuracy and training efficiency on relatively limited datasets. This study demonstrates that YOLOv8 is effective for waste classification using a transfer learning approach.

Research conducted by Mustapha *et al*. evaluated the performance of YOLOv11 in detecting biodegradable and non-biodegradable kitchen waste in real-time [6]. The system was tested under real-time scenarios to assess model accuracy and inference speed. The results showed that YOLOv11 achieved high accuracy with processing speeds suitable for real-time applications. These findings indicate an improvement in YOLOv11 performance compared to previous versions in the context of waste detection.

Recent studies have also conducted comparative evaluations among various YOLO architectures to identify performance improvements across versions. Dipo *et al*. compared YOLOv8, YOLOv9, YOLOv10, YOLOv11, and YOLOv12 using F1-score and mAP@0.5 metrics across 50 and 100 training epochs. The results demonstrated that YOLOv12 achieved the highest performance, reaching an F1-score of 0.75 and mAP@0.5 of 0.78 at 100 epochs, outperforming previous YOLO versions. In contrast, YOLOv11 exhibited comparable performance but showed limitations in detecting small objects, while YOLOv8 and YOLOv10 demonstrated stable yet slightly lower detection accuracy [7]. These findings indicate that YOLOv12 provides improved learning capability and generalization performance compared to earlier YOLO architectures.

In addition to comparisons among YOLO versions, several studies have evaluated YOLO against other object detection architectures. Keylabs reported that YOLOv8 achieved an inference latency of 1.3 ms with mAP@0.5 of 0.62, significantly outperforming Faster R-CNN, which recorded a latency of 54 ms and mAP@0.5 of 0.41 [8]. Furthermore, Srivastava *et al*. showed that YOLO consistently outperformed SSD and Faster R-CNN in real-time detection scenarios, particularly in terms of inference speed [9]. Other studies also confirmed that YOLO-based models provide superior accuracy-speed trade-offs compared to SSD, Faster R-CNN, and EfficientDet across various object detection environments. These results highlight the suitability of YOLO architectures for real-time and web-based detection systems.

Based on these comparative studies, YOLOv12 was selected as the core detection model in this research due to its superior detection accuracy, improved multi-scale object detection capability, and favorable accuracy-speed trade-off for real-time applications. Compared to earlier YOLO versions and other object detection architectures, YOLOv12 demonstrates stronger performance in detecting small and diverse objects, which is particularly important for waste detection scenarios. Unlike previous studies that primarily focused on earlier YOLO versions or non-web-based implementations, this study applies YOLOv12 within a real-time web-based waste detection system, thereby extending the application of the latest YOLO architecture in the waste management domain.

Despite the demonstrated advantages of YOLOv12 and other YOLO-based models, numerous studies on waste detection still predominantly focus on industrial environments or controlled experimental settings, with limited emphasis on public-oriented applications. In addition, performance evaluations are often conducted independently for each YOLO version, without systematic comparison under consistent conditions. Moreover, the application of YOLOv12 as the latest version, which introduces improvements in accuracy and multi-scale detection [10], has not been sufficiently explored, particularly within web-based systems designed for education and daily use by the general public.

Based on these research gaps, this study develops a web-based waste detection system that integrates a YOLOv12 model using a transfer learning approach. Transfer learning leverages pre-trained weights from large-scale datasets by retaining early convolutional layers that capture general visual features, such as edges and textures, while fine-tuning the later layers to learn task-specific features from the waste dataset [11]. This approach reduces training time and computational requirements while improving model performance, particularly when the available dataset is limited.

The system is designed to be easily accessible to the general public through various devices, helping users recognize waste categories and perform accurate waste sorting at the source. Users can perform detection by uploading waste images or activating a camera for real-time detection, after which the system displays classification results into organic, inorganic, or hazardous (B3) categories along with their subcategories. In addition, the system provides confidence scores and action recommendations for each detected waste category to support informed decision-making. This integration aims to enhance user awareness and encourage proper waste management practices in daily activities.

II. METHOD

This study was conducted by following a series of sequential processes, meaning that each stage must be completed before proceeding to the next stage. The research workflow is illustrated in Figure 1.

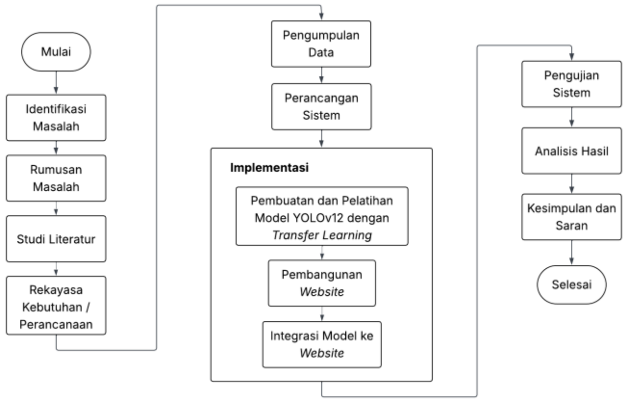

Figure 1. Research Flow

A. Problem Identification

The research begins with the process of identifying problems related to the low level of public understanding in distinguishing waste categories (organic, inorganic, and hazardous/B3). This condition causes the waste sorting process to be suboptimal, resulting in mixed waste and increasing the management burden at Final Disposal Sites. This stage is conducted to establish a basis for determining the focus and scope of the research.

B. Problem Formulation

This stage is carried out to clarify the direction of the research. The problem formulations that become the focus of this study are how to design a YOLOv12-based waste detection model and how to integrate it into a web-based system, so that it can help the public recognize types of waste quickly. These problem formulations then serve as guidelines in defining the research objectives and methodology.

C. Literature Review

At this stage, a review of the literature is conducted covering deep learning–based object detection, the development of YOLO models, transfer learning techniques, and related previous studies. The literature review is performed to strengthen the theoretical foundation, understand prior technological developments, identify research gaps, and determine the most appropriate methods to be applied.

D. Requirements Engineering/Planning

Requirements engineering is conducted to define the research requirements, which include user requirements, functional requirements, and non-functional requirements of the system.

1) *User Requirements*: User requirements are derived from the perspective of end users. In this study, user requirements include the ability to perform waste detection by uploading images or activating the camera directly (real-time), obtaining fast and accurate waste classification results, receiving information about appropriate follow-up actions for the detected waste, and accessing information about the nearest waste bank locations.

2) *Functional Requirements:* Functional requirements relate to the functions within the system. These include the system's ability to accept uploaded waste images, detect waste in real-time through a camera, perform classification using YOLOv12, display detection and classification results, provide appropriate action recommendations, and display an interactive map showing waste bank locations.

3) *Non-Functional Requirements:* Non-functional requirements are not directly related to system functions but refer to the characteristics and quality attributes required for proper system operation. These include the system being accessible at any time via a web browser, being comfortable and easy to use for users from all backgrounds, being accessible across various devices such as laptops, smartphones, and tablets, performing detection in less than 3 seconds, having a simple and intuitive user interface, and being easy to modify.

E. Data Collection

At this stage, a literature review is conducted on deep learning–based object detection, the development of YOLO models, transfer learning techniques, and similar previous studies. This process aims to strengthen the theoretical foundation, understand prior technological developments, identify research gaps, and determine the most suitable methods to be applied.

F. System Design

The system design consists of two main components: the YOLOv12 model training process and the development of a web-based system using the Waterfall method. These two processes are designed to ensure that the model is capable of accurately detecting waste categories and can be optimally integrated into the system.

1) *Model Training Process*: The YOLOv12 model training is conducted through a series of structured stages, starting from dataset preparation to model performance evaluation. The YOLOv12 training workflow is illustrated in Figure 2.

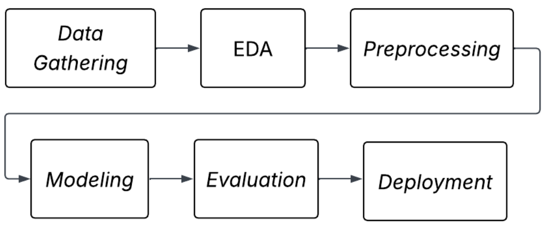

Figure 2. Model Training Workflow

The dataset used in this study consists of 2500 waste images, collected directly from the surrounding environment to represent real-world waste disposal conditions. The dataset covers 20 waste subcategories, with 125 images per subcategory, ensuring a balanced distribution at the image level. The images were captured under diverse lighting conditions, including indoor and outdoor environments, natural and artificial lighting, as well as variations in brightness and shadows. Additionally, images were taken from different viewpoints and backgrounds to increase data diversity and improve model generalization.

All images were manually annotated using the Roboflow platform, following a consistent annotation standard. Each waste object in an image was labeled using tight bounding boxes that fully enclosed the visible object area, along with its corresponding waste category label (organic, inorganic, or hazardous/B3) and subcategory. Images containing multiple waste objects were annotated with multiple bounding boxes, resulting in variations in the number of bounding boxes per image.

After annotation, an Exploratory Data Analysis (EDA) was conducted to analyze class distribution, bounding box frequency, and object diversity. The dataset was then split into training, validation, and testing sets with a ratio of 70:20:10, ensuring that each subset maintained a similar class distribution.

During the preprocessing stage, images were resized to 640 × 640 pixels, and data augmentation techniques were applied to the training set, including rotation, horizontal and vertical flipping, brightness adjustment, Gaussian blur, and motion blur. These augmentations were performed to enhance data variability and better represent real-world conditions.

The YOLOv12 model is initialized using pretrained weights through a transfer learning approach. The training process is carried out by adjusting parameters such as learning rate, batch size, number of epochs, and other relevant hyperparameters. After training is completed, the model is evaluated using precision, recall, mean Average Precision (mAP), and FPS (frames per second) metrics.

Precision is a metric that measures how often the model makes correct predictions [12]. This metric is used to assess how accurate the model is in making positive predictions. The precision formula is shown in Equation (1):

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

With

TP: positive predictions that match the ground truth labels (True Positive)

FP: positive predictions that do not match the ground truth labels (False Positive)

Recall is a metric that measures how well the model identifies all actual positive instances. This metric describes the model's ability to correctly predict positive data. The recall formula is shown in Equation (2):

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

With

TP: positive predictions that match the ground truth labels (True Positive)

FN: negative predictions that differ from the actual positive ground truth labels (False Negative)

Mean Average Precision (mAP) is the average of all AP values calculated for each class in the dataset. AP (Average Precision) is an evaluation metric that combines precision and recall into a single value and is used to evaluate bounding box localization and confidence scores. The mAP formula is shown in Equation (3):

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i \quad (3)$$

With

AP_i: AP value of the i-th class

C: total number of classes in the dataset

After the evaluation process shows satisfactory and expected results, the best-trained model is then deployed directly into the web-based system.

2) *Web System Development:* The selection of a system development method must be aligned with project requirements. Therefore, a well-structured and well-documented framework is required to avoid ad hoc development and to maintain system consistency and quality [13]. Based on these considerations, the waste detection system is developed using the Waterfall method, which is a sequential software development approach in which each stage must be completed before proceeding to the next stage [14]. The stages of the Waterfall method are illustrated in Figure 3.

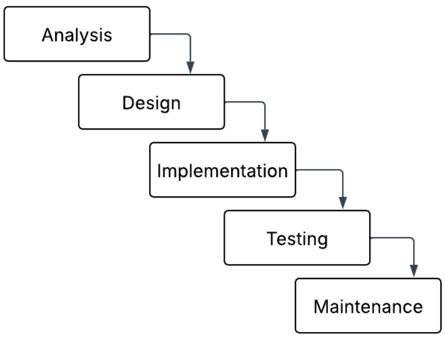

Figure 3. Stages of the Waterfall Method

The development process begins with a system requirements analysis to identify specifications that meet user needs. This analysis is conducted through brainstorming sessions and interviews with relevant stakeholders. Based on the results of this analysis, requirements modeling is performed using a use case diagram to illustrate the interactions between users and the system [15], as shown in Figure 4.

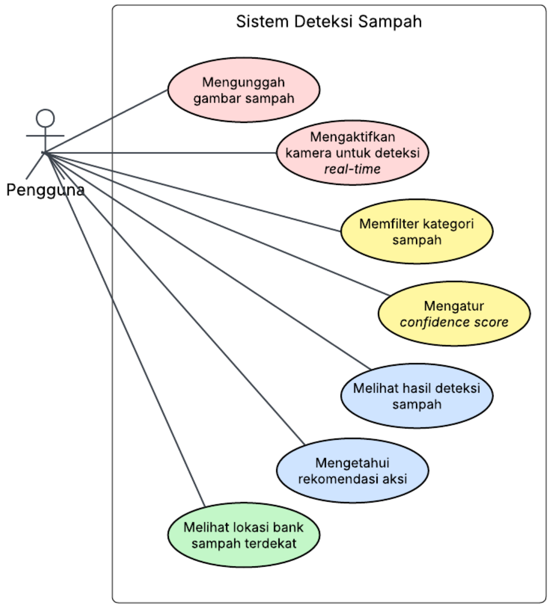

Figure 4. Use Case Diagram of the Waste Detection System

In the developed system, user interactions are modeled through several main use cases, namely uploading images or activating the camera to perform detection. The system provides options for filtering waste categories and adjusting the confidence score threshold so that the displayed detection results meet user needs. After the inference process is performed by the model, the system displays detection results in the form of bounding boxes, category labels, confidence scores, and action recommendations based on the detected waste category. Users also can obtain information about the nearest waste bank locations by granting location access to the system.

In addition to the use case diagram, requirements modeling is also conducted using an activity diagram to describe the main activity flow in the waste detection process. The activity diagram is shown in Figure 5.

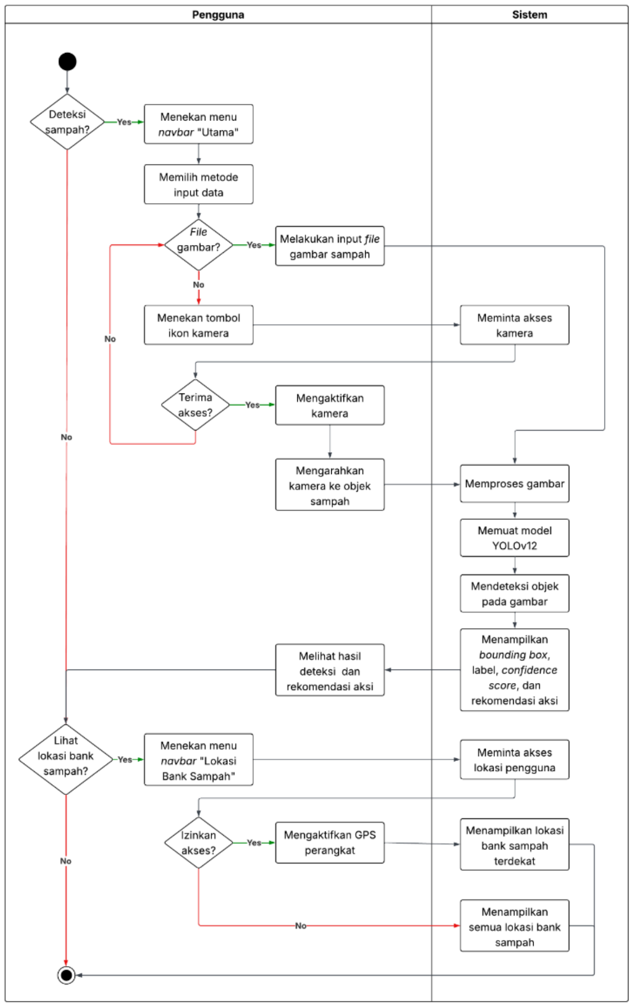

Figure 5. Activity Diagram of the Waste Detection System

In this system, the user activity flow begins with selecting the desired function, either performing waste detection or viewing waste bank location information. For the detection process, users can choose to upload an image or use the camera to obtain real-time images. The received image is then sent to the YOLOv12 model for waste detection and classification. The detection results are displayed in the form of bounding boxes, category labels, confidence scores, and action recommendations. If users want to access information about waste bank locations, the system displays the nearest locations based on the user's granted location access. However, if the user denies location access, the system displays all available waste bank locations.

A sequence diagram is also constructed to illustrate the sequence of communication between the front-end, back-end, and the YOLOv12 model during the detection process [16]. The sequence diagram is shown in Figure 6.

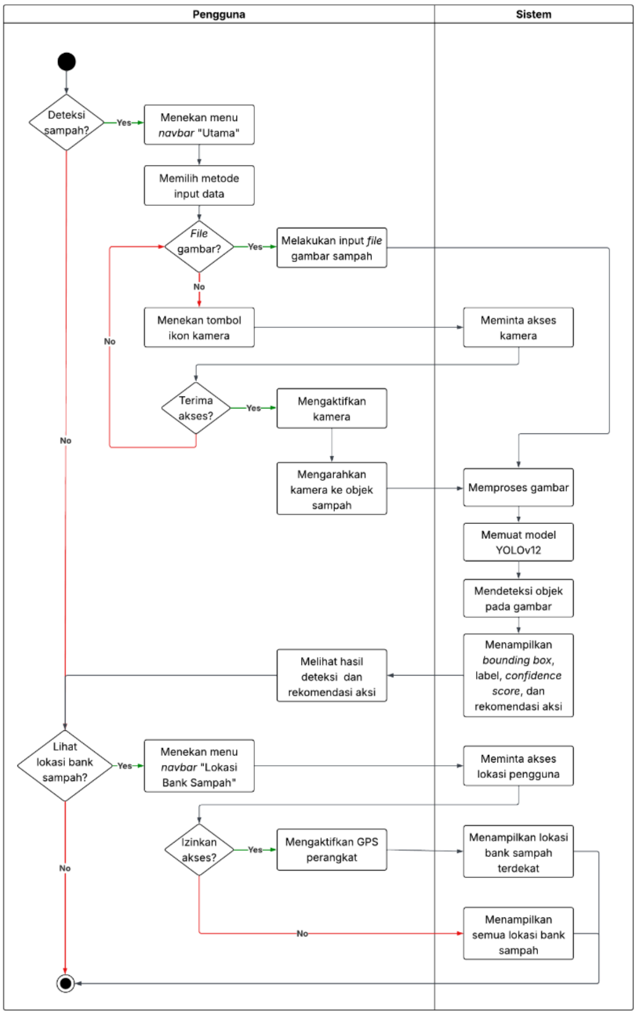

Figure 6. Sequence Diagram of the Waste Detection System

In the detection process, users can upload images or activate the camera to obtain real-time visual input. The received image is then sent to the back-end and processed by the YOLOv12 model to generate predictions in the form of bounding boxes, category labels, confidence scores, and action recommendations, which are subsequently returned and displayed to the user. For the waste bank location feature, the system obtains the user's coordinates via the device's GPS and forwards them to the back-end to search for the nearest waste bank locations. The identified locations are then sent back to the web system and presented to the user through the interface.

After the analysis stage is completed, the system development process proceeds to the design stage. The system design is carried out by ensuring a simple and intuitive user experience. This approach aligns with user-centered design principles, where the system must be designed to meet user needs and ensure user comfort so that the resulting solution is effective [17]. The design stage is conducted based on the results of the requirements analysis from the previous stage. The designed web system consists of two pages, namely the Main page and the Waste Bank Location page, with a simple layout to ensure ease of use for users from diverse backgrounds. Figure 7 shows the interface design of the main page displayed after users access the website.

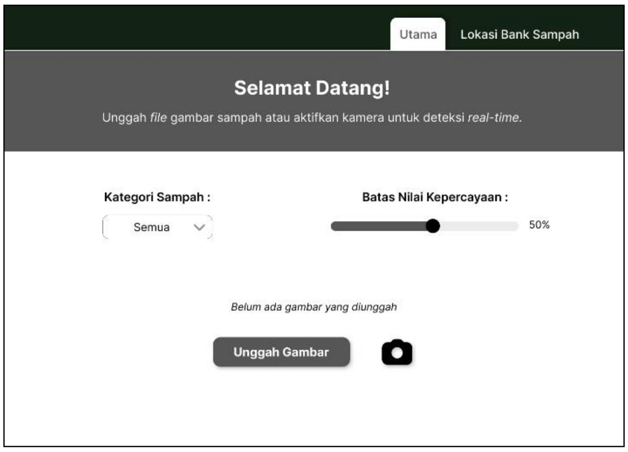

Figure 7. Main Page of the System

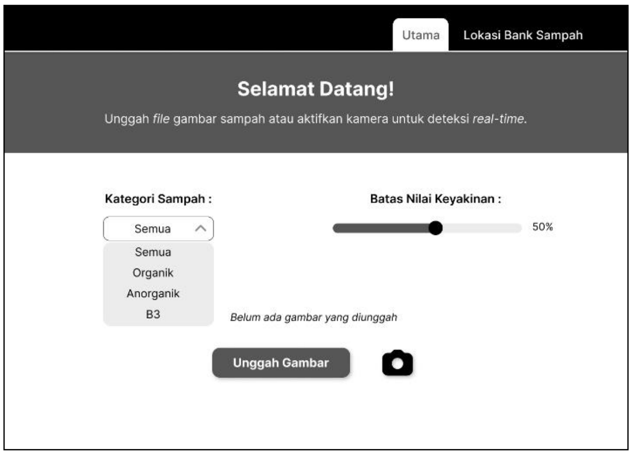

Figure 8. Interface After Dropdown Interaction

Figure 8 shows the interface when users select the waste category filter dropdown, where waste category options are displayed. The confidence score threshold filter uses a slider element, in which the percentage value changes as users move the slider thumb.

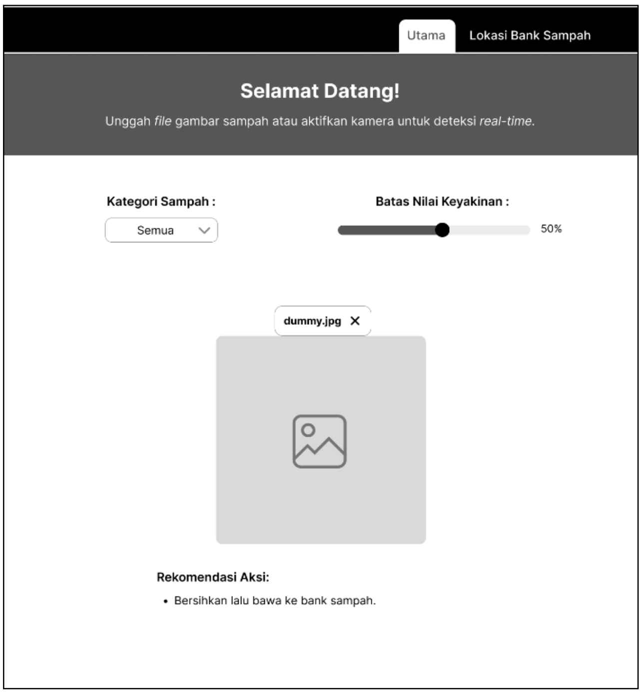

Figure 9. Interface After Image Upload

Figure 9 illustrates the interface after an image is uploaded. The uploaded image is displayed along with the detection results, including bounding boxes, category labels, confidence scores, and action recommendations.

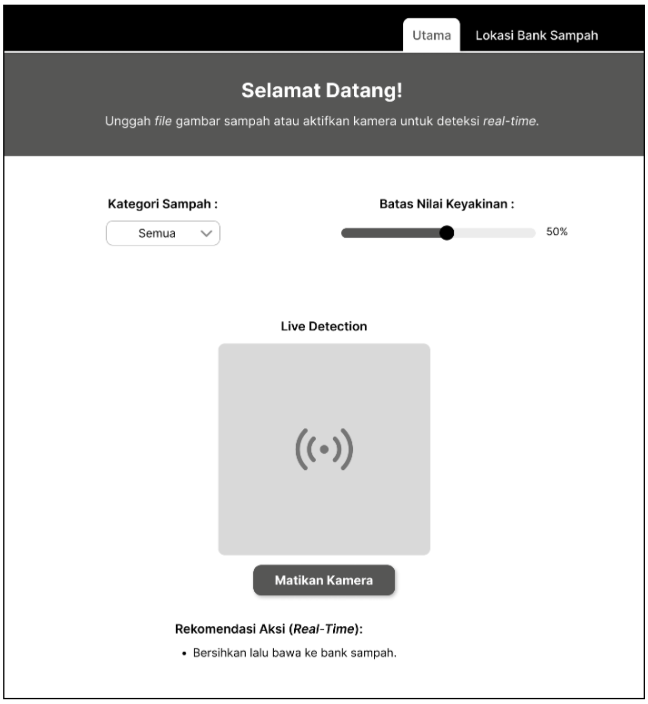

Figure 10. Camera-Based Detection Interface

Figure 10 illustrates the interface when users perform real-time detection using the camera, while Figure 11 shows the waste bank location page, which contains an interactive map.

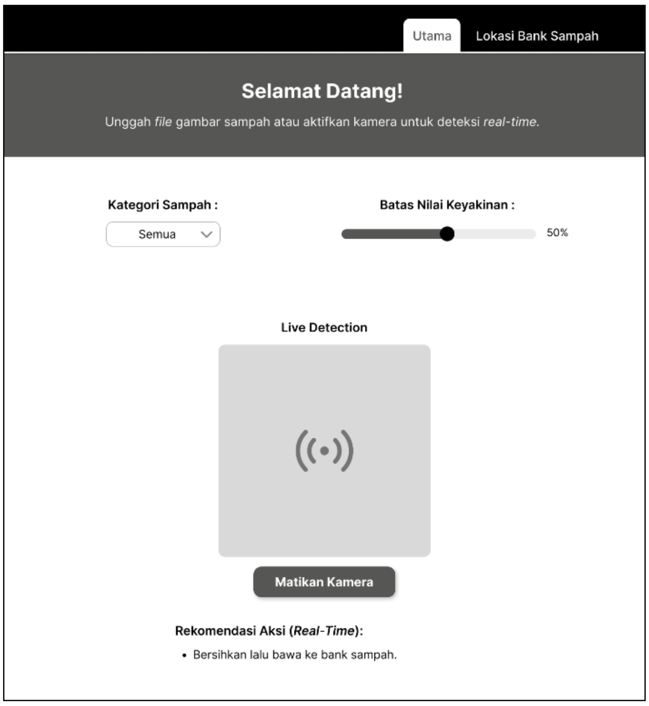

Figure 11. Waste Bank Location Page

After the design stage is completed, the resulting website interface designs are implemented into program units, forming a website that is ready for testing. During the testing stage, each unit developed in the previous stage is tested to ensure that all functionalities work properly. After unit testing, all components are integrated into a single system and tested again to ensure there are no failures. The final stage of the development process is maintenance. At this stage, the website has been fully developed and is ready for deployment. If any errors are identified that were not detected in earlier stages, the necessary fixes are treated as new system requirements.

G. Implementation

At the implementation stage, three main processes are carried out as follows.

1) *Model Training Process*: The YOLOv12 model training is conducted through a series of structured stages, starting from dataset preparation to model performance evaluation. The YOLOv12 training workflow is illustrated in Figure 2.

2) *Website Development*: The website is developed as the medium for running the detection process. It is designed with a simple and intuitive interface. The development process follows the stages of the Waterfall method illustrated in Figure 3.

3) *Model Integration into the Website*: The best-performing YOLOv12 model is integrated into the website back-end. This integration enables the model to process user-

submitted images and display prediction results directly, without requiring additional installations.

H. System Testing

The developed system, integrated with the YOLOv12 model, is then tested using black-box testing and website detection system testing. This stage is conducted to ensure that the model correctly detects waste categories, the website displays results without errors or failures, and real-time detection performance operates smoothly.

I. Result Analysis

The test results are analyzed to evaluate the effectiveness of the proposed method. The analysis uses precision, recall, mAP, and FPS metrics to determine the accuracy of the YOLOv12 model in recognizing waste categories, as well as to assess the stability and performance of the web-based system.

J. Conclusions and Recommendations

This stage represents the final phase of the research workflow. At this stage, the authors summarize the research findings, evaluate the achievement of the research objectives, and provide recommendations for future improvements and further research.

III. Result and Discussion

A. Results

The initial dataset consisted of 2500 waste images with a balanced distribution across each subcategory, namely 125 images per subcategory. All images were annotated using bounding boxes and waste category labels through the Roboflow platform. The images were then analyzed during the Exploratory Data Analysis (EDA) stage to ensure object diversity. The EDA results shown in Figure 12 indicate that the *Kulit Kuaci* (Sunflower Seed Shell) class accounts for 54.6% of the total bounding boxes, while other classes have proportions that are 1-4% lower. This finding indicates that categories with the same number of images may contain different numbers of objects within those images.

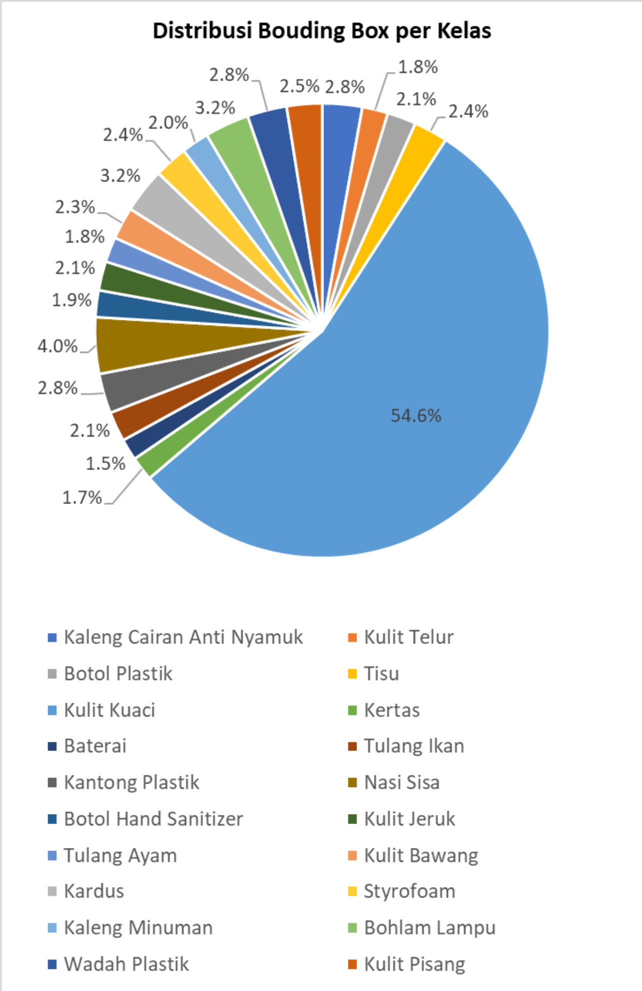

Figure 12. Bounding Box Distribution for Each Class

The preprocessing stage included resizing images to 640×640 pixels and applying data augmentation. The augmentation techniques used were horizontal and vertical flipping, rotation (−15° to +15°), brightness adjustment (−15% to +15%), exposure adjustment (−10% to +10%), blur (up to 1.5 px), and noise addition (up to 0.1% of pixels). This augmentation process increased the number of images from 2500 to 6020. The dataset was then split into training, validation, and testing sets with a ratio of 70:20:10.

The YOLOv12 model was trained using a transfer learning approach with pretrained weights from YOLOv8n, which had been trained on the COCO dataset. The YOLOv12 architecture was imported via the "yolo12.yaml" file, which defines the backbone and head structures of the object detection model. Initial training was conducted for 80 epochs by freezing several early layers, using the AdamW optimizer with a learning rate of 0.0001, a weight decay of 0.0005, a batch size of 16, an image size of 640×640 pixels, and patience = 10 for early stopping, where training stops if no performance improvement is observed over the last 10 epochs.

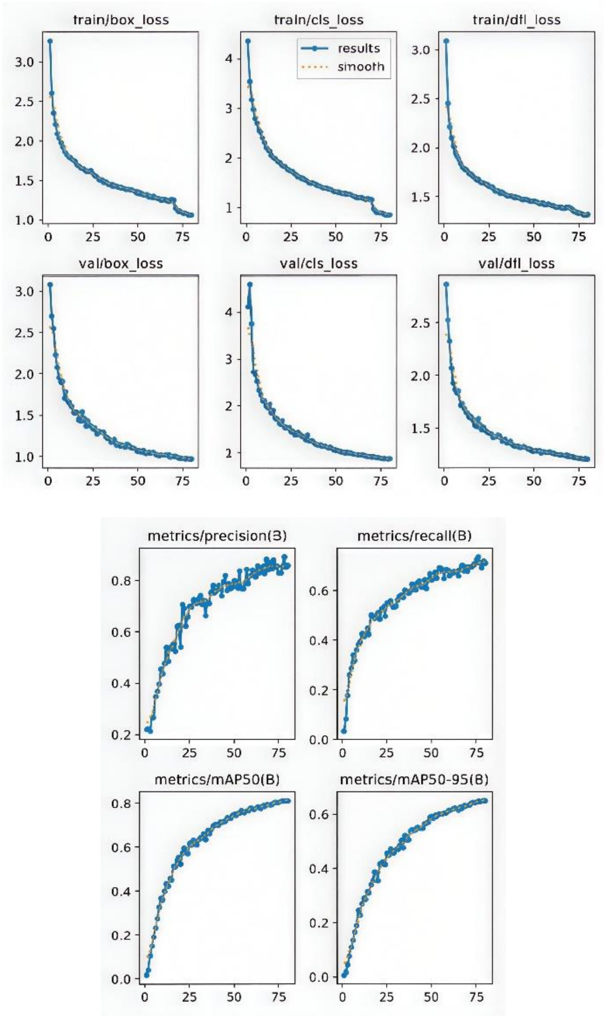

Figure 13. Initial Training Results

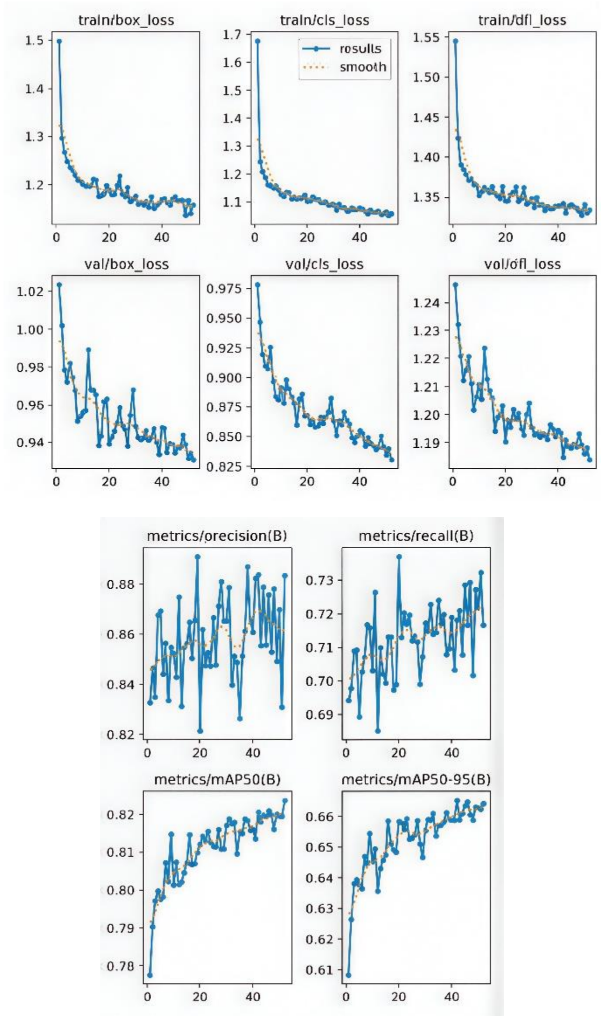

Figure 14. Fine-Tuning Results

The graphs in Figure 13 show a decrease in both training loss and validation loss, including box_loss, cls_loss, and dfl_loss, indicating that the learning process proceeded in a stable manner. Meanwhile, evaluation metrics such as precision, recall, and mAP continuously increased until convergence. At the end of training, the model achieved a precision of 0.86, a recall of 0.71, an mAP@0.5 of 0.81, and an mAP@0.5:0.95 of 0.65. These results indicate that the model learned effectively without experiencing overfitting.

Training was then continued with fine-tuning on the best model obtained from the initial training phase. Fine-tuning was performed for up to 100 epochs using the SGD optimizer, with a learning rate of 0.005, momentum of 0.937, weight decay of 0.0005, and a patience value of 10. Several light augmentation techniques were also applied, including HSV adjustments (hue 0.015, saturation 0.7, value 0.4), image translation (up to 10%), random scaling (up to 50%), and horizontal flipping with a probability of 0.5.

Figure 14 shows a more stable loss reduction and convergence compared to the initial training phase. Training stopped at epoch 52 due to early stopping, indicating that no significant performance improvement was observed over the last 10 epochs. Metrics such as precision, recall, and mAP also showed consistent improvement throughout the training process. At the final stage, the model achieved a precision of 0.88, a recall of 0.72, an mAP@0.5 of 0.82, and an mAP@0.5:0.95 of 0.66. These results demonstrate that fine-tuning successfully improved detection accuracy without introducing overfitting, allowing the model to achieve good generalization on unseen data.

After training, evaluation was conducted using the validation set. The mAP@0.5 and mAP@0.5:0.95 values presented in Table I indicate that the YOLOv12 model achieved stable performance. Precision and recall metrics were also well balanced, showing that the model can accurately recognize waste objects without overfitting. In addition, inference speed testing showed that the model processed images with an average time of 0.0180 seconds per image, equivalent to 55.67 FPS, making it suitable for real-time detection.

TABEL I
EVALUATION RESULTS ON THE VALIDATION DATASET

Metric	Value
Precision	0.8838
Recall	0.7186
mAP@0.5	0.8215
mAP@0.5:0.95	0.6649
FPS	55.67
Average Inference Time	0.0180s

Testing on the test dataset showed that the YOLOv12 model maintained consistent performance, with precision, recall, and mAP values as presented in Table 2. The model also demonstrated high inference speed, with an average processing time of 0.0187 seconds per image or 53.40 FPS, indicating that the system remains responsive to new data. These results confirm that the model can detect waste objects quickly and accurately under various testing conditions and effectively supports real-time detection.

TABEL II
EVALUATION RESULTS ON THE TEST DATASET

Metric	Value
Precision	0.8602
Recall	0.7362
mAP@0.5	0.8287
mAP@0.5:0.95	0.6788
FPS	53.40
Average Inference Time	0.0187s

B. System Development Results

The web-based system was developed using HTML, CSS, JavaScript, and the Flask framework as the back-end. Flask was chosen due to its lightweight and flexible nature, as well as its minimal dependencies, which simplify website development [18]. To support system accessibility without requiring additional installation, the waste detection application was deployed on the Hugging Face Spaces platform. This platform provides a lightweight hosting environment, supports Flask integration, and allows the model to be executed directly through a web browser.

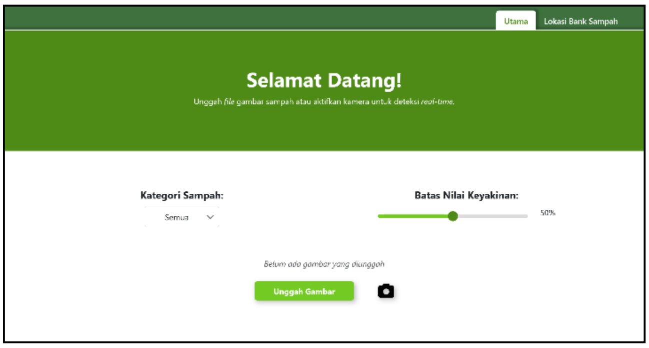

Figure 15. Main Page

Figure 15 shows the page displayed when the website is first accessed. This page serves as the Main page where waste detection is performed.

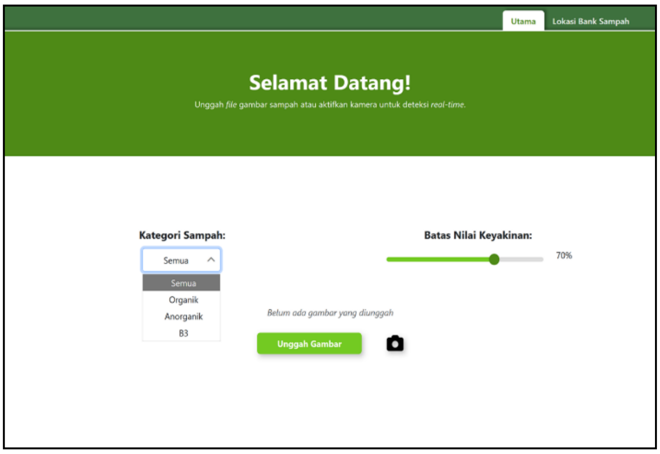

Figure 16. Detection Filters

Users can configure detection filters according to their preferences. The category filter allows users to select a specific waste category to be detected, while the confidence threshold filter indicates the level of confidence required by the model in recognizing waste objects.

Figure 17. Detection via Image Upload

Figure 17 shows the interface after users upload an image for detection. The uploaded image file name is displayed along with a red cross (×) icon that allows users to remove the file and perform detection again. An image preview is displayed together with the detection results and action recommendations.

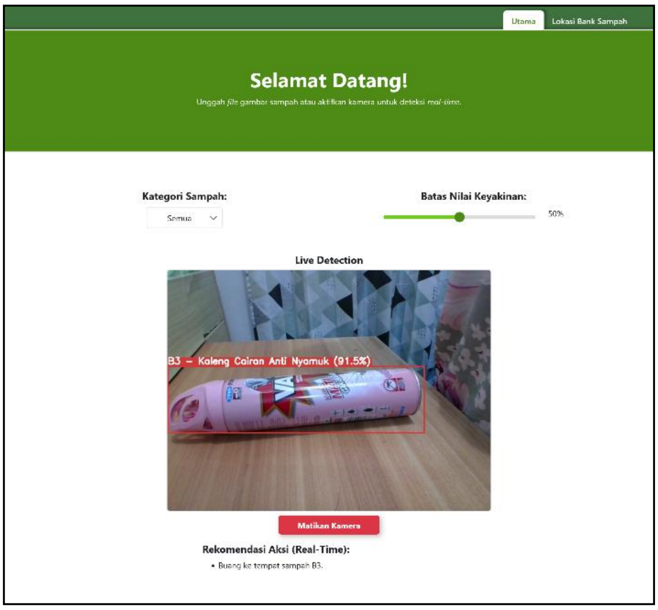

Figure 18. Real-Time Detection Using Camera

Figure 18 shows the real-time detection interface when camera access is activated and permitted. Detection results and action recommendations are displayed in real time according to the objects captured by the camera.

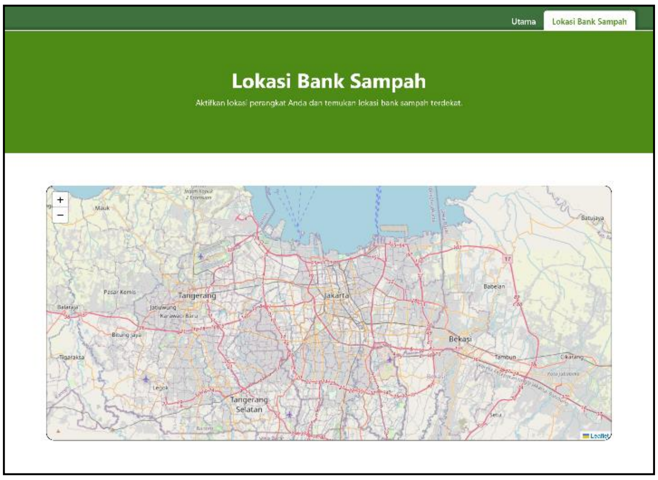

Figure 19. Waste Bank Location Page

Figure 19 shows the initial view of the Waste Bank Location page, where the website requests permission to access the user's location. If permission is granted, the nearest waste bank locations are displayed along with the user's current location point, as shown in Figure 20. However, if the user denies location access, all available waste bank locations are displayed on the map, as shown in Figure 21.

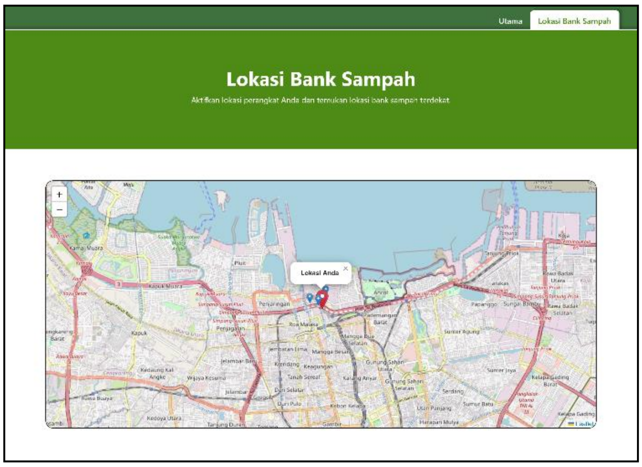

Figure 20. Nearest Waste Bank Locations

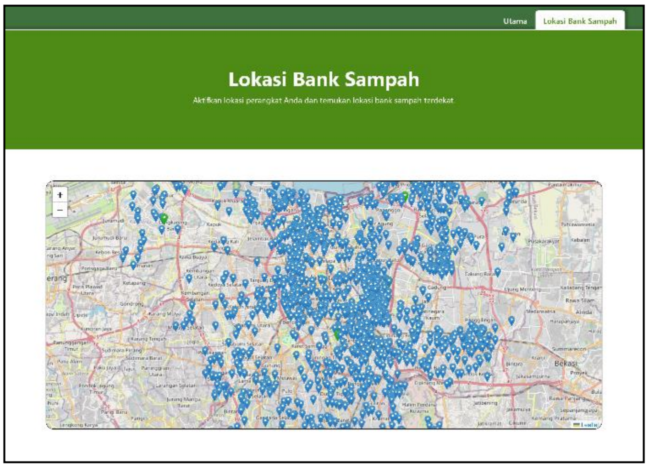

Figure 21. All Waste Bank Locations

C. *System Testing*

The system testing conducted includes black-box testing and detection accuracy testing, as described below.

1) *Black-Box Testing*: Black-box testing is a software testing method used to evaluate system functionality without examining the internal code or structure [19]. The testing results are presented in Table 3.

TABEL III
BLACK-BOX TESTING

No.	Test Case	Expected Result	Test Result
1.	Navigation menu functionality	The displayed page matches the active navigation menu.	Successful
2.	Waste category filter	The category changes according to user selection.	Successful
3.	Confidence threshold filter	The confidence value changes according to the	Successful

		slider thumb position.	
4.	Image upload	File name, image preview, detection results, and action recommendations are displayed.	Successful
5.	Real-time detection with camera access	Real-time preview, detection results, action recommendations, and the “Turn Off Camera” button are displayed.	Successful
6.	Turn off camera	The camera is turn off and the real-time preview disappears.	Successful
7.	Map display	The map displays waste bank locations.	Successful
8.	Website responsiveness	The website can be accessed on various devices and adapts to different screen sizes.	Successful

2) *Detection Accuracy Testing:* Detection accuracy testing is conducted using two approaches, including image upload-based detection and real-time detection. The results are presented in Table 4 and Table 5.

TABEL IV
DETECTION RESULTS USING IMAGE UPLOAD

Detection Result	Ground Truth	Confidence Score
Organic – Banana Peel	Organic – Banana Peel	95.3%
Organic – Orange Peel	Organic – Orange Peel	97.4%
Organic – Leftover Rice	Organic – Leftover Rice	91.4%
Organic – Chicken Bone	Organic – Chicken Bone	91.5%
Organic – Fish Bone	Organic – Fish Bone	83.0%
Organic – Eggshell	Organic – Eggshell	94.9%
Organic – Onion Peel	Organic – Onion Peel	74.3%
Organic – Sunflower Seed Shell	Organic – Sunflower Seed Shell	82.9%
Inorganic – Plastic Bottle	Inorganic – Plastic Bottle	94.2%
Inorganic – Plastic Bag	Inorganic – Plastic Bag	95.1%
Inorganic – Plastic Container	Inorganic – Plastic Container	93.7%
Inorganic – Beverage Can	Inorganic – Beverage Can	88.1%
Inorganic – Cardboard	Inorganic – Cardboard	93.2%
Inorganic – Paper	Inorganic – Paper	96.1%
Inorganic – Tissue	Inorganic – Tissue	93.8%
Inorganic – Styrofoam	Inorganic – Styrofoam	95.9%
B3 – Battery	B3 – Battery	93.3%
B3 – Light Bulb	B3 – Light Bulb	92.5%
B3 – Mosquito Repellent Can	B3 – Mosquito Repellent Can	82.2%
B3 – Hand Sanitizer Bottle	B3 – Hand Sanitizer Bottle	88.8%

TABEL V
REAL-TIME DETECTION RESULTS

Detection Result	Ground Truth	Confidence Score
Organic – Banana Peel	Organic – Banana Peel	95.5%
Organic – Orange Peel	Organic – Orange Peel	70.7%
Organic – Leftover Rice	Organic – Leftover Rice	96.4%
Organic – Chicken Bone	Organic – Chicken Bone	94.5%
Organic – Fish Bone	Organic – Fish Bone	85.1%
Organic – Eggshell	Organic – Eggshell	93.1%
Organic – Onion Peel	Organic – Onion Peel	94.0%
Organic – Sunflower Seed Shell	Organic – Sunflower Seed Shell	87.1%
Inorganic – Plastic Bottle	Inorganic – Plastic Bottle	74.3%
Inorganic – Plastic Bag	Inorganic – Plastic Bag	88.7%
Inorganic – Plastic Container	Inorganic – Plastic Container	91.0%
B3 – Battery	Inorganic – Beverage Can	89.1%
Inorganic – Cardboard	Inorganic – Cardboard	90.9%
Inorganic – Paper	Inorganic – Paper	89.0%
Inorganic – Tissue	Inorganic – Tissue	80.3%
Inorganic – Styrofoam	Inorganic – Styrofoam	94.2%
B3 – Battery	B3 – Battery	94.9%
B3 – Light Bulb	B3 – Light Bulb	93.6%
B3 – Mosquito Repellent Can	B3 – Mosquito Repellent Can	81.0%
B3 – Hand Sanitizer Bottle	B3 – Hand Sanitizer Bottle	86.8%

The testing results indicate that the system is capable of detecting and classifying waste objects with high accuracy. In the image upload-based testing, all samples were correctly identified according to the ground truth labels, with confidence scores ranging from 74.3% to 97.4%. Meanwhile, real-time camera-based testing also demonstrated good performance, with confidence scores ranging from 70.7% to 95.5%.

Despite the overall high detection accuracy, several limitations were observed during real-time testing. The most notable error occurred when a beverage can was misclassified as a battery, which represents a false positive for the B3 category and a false negative for the inorganic category. This type of error indicates confusion between subcategories that share similar visual characteristics, particularly cylindrical shape, metallic appearance, and reflective surfaces.

In addition to misclassification, certain classes exhibited lower confidence scores, such as onion peel, sunflower seed shell, and plastic bottle during real-time detection. These cases are primarily influenced by lighting variations, motion blur, and partial occlusion, which commonly occur in real-world usage scenarios. Objects with small size, irregular texture, or deformable shapes were more susceptible to reduced confidence, even when correctly classified.

These findings suggest that while YOLOv12 demonstrates strong generalization capability, its performance can be affected by visual similarity across waste subcategories and challenging real-time conditions. Future improvements may include incorporating additional training samples for visually similar classes, applying class-aware data augmentation, and integrating post-detection filtering or hierarchical classification to reduce confusion between hazardous (B3) and non-hazardous waste categories.

While the proposed system demonstrates strong performance on the collected dataset, its generalization to unseen waste categories and new environmental conditions remains a challenge. The current model relies on supervised learning and therefore requires additional annotated data to adapt to new waste types or changes in visual appearance caused by different lighting, backgrounds, or camera devices.

From a sustainability perspective, the system can be extended through incremental learning or periodic model retraining using newly collected data, allowing it to continuously adapt to evolving waste patterns. In addition, the modular design of the web-based system enables future integration of updated models without major changes to the application layer. These strategies can support the long-term usability and deployment of the system beyond the initial data collection environment.

Although the proposed system provides waste classification results, confidence scores, and action recommendations, this study did not directly evaluate changes in user behavior, such as increased awareness or participation in waste sorting. However, the inclusion of explanatory feedback and information on waste handling and waste banks has the potential to support user understanding and decision-making, particularly for users with limited prior knowledge of waste categories. Future studies may incorporate user-based evaluations to quantitatively assess the system's impact on user awareness and waste sorting behaviour.

IV. Conclusion

This study successfully developed a web-based waste detection system using the YOLOv12 model through a transfer learning approach. Compared to previous studies that mainly focused on offline or mobile-based detection, the proposed system extends existing findings by enabling real-time waste detection directly through a web platform without requiring additional installation. The experimental results demonstrate that transfer learning and fine-tuning effectively improve both detection accuracy and inference speed, achieving a precision of 0.86, recall of 0.74, mAP@0.5 of 0.83, mAP@0.5-0.95 of 0.67, and an inference speed of 53.40 FPS.

System testing also shows a high level of consistency between detection results and ground truth labels, with an average confidence score above 80% under various testing scenarios. These results indicate that the proposed system has good generalization capability and can operate reliably under diverse lighting conditions, backgrounds, and image qualities in real-world environments.

Although the results are promising, this study is still limited by the size and diversity of the dataset and the occurrence of misclassification in visually similar waste subcategories, particularly under real-time conditions. Therefore, future research may focus on improving detection robustness through dataset expansion, class-aware augmentation, and hierarchical classification, as well as exploring adaptive learning strategies to support generalization to unseen waste categories and evolving environmental conditions. Beyond technical enhancements, future work should also incorporate user-centered evaluations to assess the impact of the proposed system on public awareness, participation, and waste sorting behavior.

In addition to model-level improvements, future work may also explore system-level integration to increase the practical impact of the proposed approach. The waste detection system can be extended to mobile applications to enable on-site waste classification using smartphone cameras, or integrated with IoT-based smart bins to support automatic waste sorting at the source. Furthermore, integration with local government or municipal waste management systems could enable real-time data collection, monitoring, and decision support for waste management policies. Such integrations have the potential to enhance the scalability and real-world applicability of the proposed system.

References

[1] Kementerian Lingkungan Hidup dan Kehutanan, "Capaian Kinerja Pengelolaan Sampah." Accessed: Jun. 20, 2025. [Online]. Available: https://sipsn.kemenlh.go.id/sipsn/

[2] E. Supriyanto, R. R. Isnanto, and S. H. Purnomo, *Deteksi Dan Klasifikasi Kesehatan Ayam Menggunakan Yolo Dengan Google Colab*. Pagar Alam: PENERBIT LD Media, 2025.

[3] O. Soerya, N. Utomo, F. Utaminingrum, and E. R. Widasari, "Implementasi YOLO versi 3 untuk Mengidentifikasi dan Mengklasifikasi Sampah Kantor berbasis NVIDIA Jetson Nano," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 6, pp. 2829–2834, Jun. 2022, [Online]. Available: http://j-ptiik.ub.ac.id

[4] H. Abdillah, A. N. Syahbana, G. I. Al Husain, and S. Agustin, "Detektif Sampah : Klasifikasi Jenis Sampah Organik dan Anorganik Menggunakan Metode YOLOv5 Berbasis Website," *Jurnal INOVATIF WIRA WACANA*, vol. 3, no. 2, pp. 128–135, Aug. 2024.

[5] N. Suwela and M. Z. Hedriyadi, "Deteksi Jenis Sampah Menggunakan Metode Transfer Learning YOLO-V8," *Jurnal Sistem Informasi dan Sains Teknologi*, vol. 7, no. 1, Feb. 2025.

[6] A. A. Mustapha, S. 'Atifah Saruchi, M. I. Solihin, F. K. Aldeen, and A. A. M. Al-Talib, "Exploring the Performance of YOLOv11: Detecting Compostable and Non-Compostable Kitchen Waste in Real-Time Applications," *The 2025 International Conference on Artificial Life and Robotics (ICAROB2025)*, pp. 13–16, Feb. 2025.

[7] M. H. Dipo *et al.*, "Real-Time Waste Detection and Classification Using YOLOv12-Based Deep Learning Model," *Digital*, vol. 5, no. 2, pp. 1–17, Jun. 2025, doi: 10.3390/digital5020019.

[8] Keylabs, "YOLOv8 vs Faster R-CNN: A Comparative Analysis," Keylabs. Accessed: Aug. 17, 2025. [Online]. Available: https://keylabs.ai/blog/yolov8-vs-faster-r-cnn-a-comparative-analysis/?utm_source=chatgpt.com

[9] S. Srivastava, A. V. Divekar, C. Anilkumar, I. Naik, V. Kulkarni, and V. Pattabiraman, "Comparative analysis of deep learning image detection algorithms," *J Big Data*, vol. 8, May 2021, doi: 10.1186/s40537-021-00434-w.

[10] Ultralytics Inc., "YOLO12: Attention-Centric Object Detection." Accessed: Jun. 21, 2025. [Online]. Available: https://docs.ultralytics.com/models/yolo12/

[11] A. E. Putra, M. F. Naufal, and V. R. Prasetyo, "Klasifikasi Jenis Rempah Menggunakan Convolutional Neural Network dan Transfer Learning," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 9, no. 1, Apr. 2023.

[12] M. W. S. Sanjaya, *Deep Learning Citra Medis Berbasis Pemrograman Python*. Bandung: BOLABOT, 2023.

[13] I. G. N. Suryantara, Michael, J. F. Andry, and J. A. Ginting, "PENGEMBANGAN APLIKASI OPERASIONAL RESTORAN DENGAN FRAMEWORK SCRUM (STUDI KASUS: RESTORAN PT.XYZ)," *Infotech: Journal of Technology Information*, vol. 9, no. 2, pp. 117–128, Nov. 2023, doi: 10.37365/jti.v9i2.168.

[14] L. Syaputri, E. G. Putra, E. Syahrani, E. Dwian, and F. Purwani, "PERBANDINGAN EFEKTIVITAS METODE WATERFALL DAN AGILE DALAM PENGEMBANGAN SISTEM INFORMASI SEBUAH SYSTEMATIC LITERATURE REVIEW," *Journal of Scientech Research and Development*, vol. 6, no. 2, pp. 262–273, 2024, doi: 10.35870/jtik.v6i1.380.

[15] S. Iqbal, I. Al-Azzoni, G. Allen, and H. U. Khan, "Extending UML Use Case Diagrams to Represent Non-Interactive Functional Requirements," *E-Informatica Software Engineering Journal*, vol. 14, no. 1, pp. 97–115, 2020, doi: 10.37190/e-INF200104.

[16] S. Nabila, A. R. Putri, A. Hafizhah, F. H. Rahmah, and R. Muslikhah, "Pemodelan Diagram UML Pada Perancangan Sistem Aplikasi Konsultasi Hewan Peliharaan Berbasis Android (Studi Kasus: Alopet)," *Jurnal Ilmu Komputer dan Bisnis*, vol. 12, no. 2, pp. 130–139, Nov. 2021, doi: 10.47927/jikb.v12i2.150.

[17] J. F. Andry, F. S. Lee, K. Christianto, Y. Purnomo, A. Chakir, and L. Liliana, "User-Centric Development of Good Delivery Applications Using Design Thinking and Business Model Canvas," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 12, no. 4, Oct. 2025, doi: 10.4108/eetsis.8223.

[18] R. S. Samosir, E. Lumba, and P. P. Situmorang, "A PROTOTYPE OF DIGITAL LIBRARY APPLICATION USING MICROFRAMEWORK FLASK," *Jurnal Techno Nusa Mandiri*, vol. 19, no. 2, pp. 96–103, Sep. 2022, doi: 10.33480/techno.v19i2.3006.

[19] N. M. D. Febriyanti, A. A. K. O. Sudana, and I. N. Piarsa, "Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen," *Jurnal Ilmiah Teknologi dan Komputer*, vol. 2, no. 3, Dec. 2021.