



Aplikasi Pendeteksi dan Pengklasifikasi Sampah Berbasis Android Menggunakan Algoritma SSD MobileNetV2

Eroldy Rumayar, Quido Conferti Kainde*, Kristofel Santa

Fakultas Teknik, Teknik Informatika, Universitas Negeri Manado, Tondano, Indonesia
Email: ¹19210011@unima.ac.id, ^{2,*}quidokainde@unima.ac.id, ³kristofelsanta@unima.ac.id
Email Penulis Korespondensi: quidokainde@unima.ac.id

Abstrak—Penelitian ini bertujuan untuk mengembangkan aplikasi pendeteksi dan pengklasifikasi sampah menggunakan algoritma SSD MobileNetV2 yang berbasis aplikasi android. Masalah mengenai sampah, terutama timbulan sampah, merupakan masalah yang sangat krusial untuk diatasi, salah satu faktornya adalah minimnya kesadaran masyarakat akan pemilahan sampah. Berbagai upaya untuk meningkatkan kesadaran masyarakat akan pemilahan sampah yang telah dilakukan seperti sosialisasi, penyuluhan, penggunaan media brosur dan media poster, membutuhkan banyak waktu, tenaga, dan sumber daya. Penelitian ini dilakukan untuk mengusulkan cara atau pendekatan yang lebih efisien dan lebih praktis terhadap edukasi sekaligus penanganan pemilahan sampah, yaitu dengan menggunakan aplikasi android yang terintegrasi dengan algoritma *object detection* SSD MobileNetV2. Metode yang digunakan dalam penelitian ini terdiri dari proses pengumpulan *dataset* objek-objek sampah dengan jenis organik, anorganik, dan B3 (bahan berbahaya dan beracun), kemudian melakukan pelatihan algoritma SSD MobileNetV2 menggunakan *framework* MediaPipe *mediapipe-model-maker*, dan pengembangan aplikasi android dengan integrasi algoritma SSD MobileNetV2 yang telah dilatih, menggunakan *framework* MediaPipe *library* Mediapipe Tasks Vision. Penelitian ini menghasilkan *dataset* buatan berformat Pascal VOC dengan jumlah 4302 gambar objek-objek sampah kemudian dibagi menggunakan rasio 80% untuk *training set* dan 20% untuk *validation set*. *Dataset* yang dibuat kemudian dilatihkan pada algoritma SSD MobileNetV2 dengan hasil performa AP IoU=0.50:0.95 dengan nilai 0.847, AP IoU=0.50 dengan nilai 0.986, dan AP IoU=0.75 dengan nilai 0.969. Algoritma SSD MobileNetV2 yang telah dilatih kemudian diintegrasikan dengan aplikasi android yang dikembangkan, hasil pengujian pada perangkat Android menengah ke atas mendapatkan waktu inferensi rata-rata berkisar antara 165-230 ms, selain itu aplikasi ini juga berhasil mendeteksi objek-objek sampah sesuai dengan yang dilatihkan pada model, aplikasi ini memiliki fitur *scan* yang berjalan secara *real-time* dengan mekanisme klasifikasi jenis objek sampah yang dideteksi menggunakan warna *bounding box*, untuk jenis sampah organik berwarna hijau, sampah anorganik berwarna kuning, dan sampah B3 berwarna merah. Dengan mekanisme ini memberikan pengalaman yang interaktif kepada pengguna dalam memilah sampahnya, sehingga diharapkan meningkatkan kesadaran akan pemilahan sampah.

Kata Kunci: Deteksi Objek; MediaPipe; Pemilahan Sampah; SSD MobileNetV2; Visi Komputer

Abstract—This study aims to develop a waste detection and classification application using the SSD MobileNetV2 algorithm based on an Android application. The problem of waste, especially waste generation, is a crucial issue that needs to be addressed, one of the factors being the lack of public awareness regarding waste sorting. Various efforts to increase public awareness of waste sorting that have been carried out, such as socialization, counseling, the use of brochure media, and poster media, require a lot of time, effort, and resources. This research was conducted to propose a more efficient and practical approach to education as well as handling waste sorting, namely by using an Android application integrated with the SSD MobileNetV2 object detection algorithm. The method used in this study consists of the process of collecting datasets of waste objects with types of organic, inorganic, and hazardous and toxic materials (B3), then training the SSD MobileNetV2 algorithm using the MediaPipe framework with the mediapipe-model-maker library, and developing an Android application integrated with the trained SSD MobileNetV2 algorithm using the MediaPipe framework with the Mediapipe Tasks Vision library. This study produced a synthetic dataset in Pascal VOC format with a total of 4302 images of waste objects divided into 80% for the training set and 20% for the validation set. The created dataset was then trained on the SSD MobileNetV2 algorithm with performance results of AP IoU=0.50:0.95 with a value of 0.847, AP IoU=0.50 with a value of 0.986, and AP IoU=0.75 with a value of 0.969. The trained SSD MobileNetV2 algorithm was then integrated into the developed Android application. The testing results on mid to high-end Android devices obtained an average inference time ranging from 165–230 ms. In addition, this application successfully detected waste objects according to those trained in the model. This application features a real-time scanning function with a classification mechanism for detected waste object types using bounding box colors, where organic waste is marked in green, inorganic waste in yellow, and B3 waste in red. With this mechanism, it provides an interactive experience for users in sorting their waste, thereby expected to increase awareness of waste sorting.

Keywords: Object Detection; MediaPipe; Waste Sorting; SSD MobileNetV2; Computer Vision

1. PENDAHULUAN

Sampah merupakan masalah yang sangat krusial untuk diatasi, berbagai penelitian untuk mengatasi permasalahan sampah telah dilakukan misalnya penelitian untuk melakukan estimasi timbulan sampah (Martina Lorensa et al., 2024), penelitian untuk melakukan *tracking* dan monitoring truk pengangkut sampah (Daeng et al., 2025), dan penelitian untuk mengembangkan aplikasi *e-commerce* untuk pengelolaan sampah (Walangitan & Rompas, 2023). Penelitian ini berfokus pada masalah pemilahan sampah.

Berdasarkan data dari Sistem Informasi Pengelolaan Sampah Nasional (SIPSN) Kementerian Lingkungan Hidup dan Kehutanan (KLHK) jumlah timbulan sampah nasional Indonesia per 2 Oktober 2024 mencapai 38,79 juta ton, dari total sampah tersebut yang bisa dikelola sebanyak 24,1 juta ton atau 62,13% dan sampah yang tidak bisa dikelola sebanyak 14,69 juta ton atau 37,87% (Kementerian Lingkungan Hidup dan Kehutanan, 2024). Salah satu faktor banyaknya sampah nasional yang tidak bisa dikelola disebabkan karena minimnya kesadaran masyarakat Indonesia akan pemilahan sampah (Humas BRIN, 2022). Berbagai upaya untuk meningkatkan kesadaran masyarakat akan pemilahan sampah telah dilakukan antara lain edukasi dengan cara sosialisasi (Khomsyi et al., 2024), penyuluhan

(Yuliawati et al., 2024), menggunakan media brosur (Rosmaini & Birman, 2022), dan menggunakan media poster (Meilani & Harianti, 2024), berbagai cara yang telah dilakukan dalam meningkatkan kesadaran masyarakat akan pemilahan sampah sangat baik, namun cara-cara tersebut membutuhkan banyak waktu, tenaga, dan sumber daya. Penelitian ini mengusulkan cara atau pendekatan yang lebih efisien dan lebih praktis terhadap edukasi sekaligus penanganan pemilahan sampah yaitu dengan menggunakan aplikasi android yang terintegrasi dengan teknologi *artificial intelligence* spesifiknya teknologi *computer vision*.

Penelitian serupa mengenai penerapan teknologi *computer vision* dalam pemilahan sampah telah dilakukan contohnya penelitian yang berjudul *Deep learning implementation using convolutional neural network in inorganic packaging waste sorting*, penelitian ini menghasilkan sistem klasifikasi sampah untuk kemasan sampah anorganik dengan arsitektur *convolutional neural network* (CNN) Densenet 201 menjadi yang terbaik dengan akurasi mencapai 95,31% pada *validation set* (Laksono et al., 2024). Selanjutnya penelitian yang berjudul *MWaste: An app that uses deep learning to manage household waste*, penelitian ini menghasilkan sistem klasifikasi sampah dengan kategori *plastic, paper, metal, glass, cardboard, trash*, yang diintegrasikan dalam aplikasi *mobile* bernama MWaste (Kunwar, 2023). Kemudian penelitian yang berjudul *Garbage classification algorithm based on improved MobileNetV3*, penelitian ini menghasilkan sistem klasifikasi sampah dengan *dataset* yang dibuat peneliti penelitian ini, menggunakan algoritma MobileNetV3 yang telah dimodifikasi dan menghasilkan akurasi 96,55% (Tian et al., 2024). Penelitian serupa dengan judul *RealWaste: A Novel Real-Life Data Set for Landfill Waste Classification Using Deep Learning*, penelitian ini melakukan analisa terhadap berbagai arsitektur algoritma CNN dalam mengklasifikasi *dataset* bernama RealWaste, berisi gambar sampah yang dikumpulkan dari TPA (tempat pembuangan akhir), menghasilkan akurasi hingga 89,19% pada arsitektur Inception V3 (Single et al., 2023). Selanjutnya penelitian dengan judul *Application research of image classification algorithm based on deep learning in household garbage sorting*, penelitian ini mengusulkan metode klasifikasi sampah rumah tangga berbasis CNN yang dioptimalkan dengan *Capuchin Search Algorithm* (CapSA) dan dikombinasikan dengan *Error-Correcting Output Codes* (ECOC) berbasis *Artificial Neural Network* (ANN), menghasilkan akurasi hingga 99,01%.

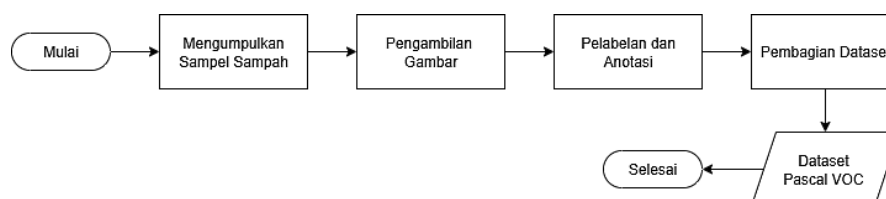
Perbedaan penelitian-penelitian sebelumnya dengan penelitian ini adalah penelitian-penelitian sebelumnya menggunakan pendekatan atau teknologi *computer vision image classification*, sedangkan penelitian ini menggunakan pendekatan atau teknologi *computer vision object detection*. Pendekatan *image classification* hanya mampu mengenali kategori sampah dari satu objek dominan dalam gambar, tanpa mengetahui lokasi atau jumlah objek yang terdeteksi. Sebaliknya, *object detection* mampu mendeteksi banyak objek sampah sekaligus beserta posisi masing-masing objek (Szeliski, 2018), *object detection* juga memiliki keunggulan dalam mendukung mekanisme pemrosesan secara *real-time* (Khan et al., 2023), sehingga lebih efektif untuk aplikasi pemilah sampah di lingkungan nyata. Penerapan *object detection* sangat relevan dalam penelitian ini karena memungkinkan sistem untuk mendeteksi setiap objek sampah secara *real-time* dan melokalisasi posisinya menggunakan *bounding box*, sekaligus mengidentifikasi jenis sampah yang terdeteksi secara visual. Salah satu algoritma dalam pengimplementasian teknologi *object detection* adalah SSD MobileNetV2, algoritma ini dibangun dengan fokus pada efisiensi yang membuatnya sangat cocok untuk perangkat *mobile* yang memiliki keterbatasan sumber daya (Sandler et al., 2018). Pemilihan perangkat *mobile* didasarkan pada dua pertimbangan utama: portabilitas dan efisiensi integrasi *hardware*, perangkat ini sudah memiliki semua komponen penting yang dibutuhkan oleh algoritma *object detection* untuk mendeteksi dan mengidentifikasi jenis sampah, yaitu kamera sebagai masukan visual, GPU untuk komputasi *real-time*, dan layar untuk menampilkan hasil deteksi. Dilansir dari situs gs.statcounter.com per agustus 2025, dari total perangkat *mobile* di Indonesia, 89,63% menggunakan sistem operasi android (Statcounter, 2025). Dengan demikian, penelitian ini menggunakan algoritma *object detection* SSD MobileNetV2 yang diintegrasikan dalam aplikasi *mobile* berbasis android sebagai solusi yang efisien dan praktis terhadap edukasi sekaligus penanganan pemilahan sampah.

2. METODOLOGI PENELITIAN

Penelitian ini terdiri dari tahapan pengumpulan dan persiapan *dataset*, pelatihan algoritma SSD MobileNetV2, dan pengembangan aplikasi android.

2.1 Pengumpulan dan Persiapan *Dataset*

Tahapan pengumpulan dan persiapan *dataset* merupakan tahapan untuk membuat *dataset* yang terdiri dari gambar-gambar objek sampah. Penelitian ini menggunakan objek-objek sampah yang terdiri dari 3 jenis sampah yaitu sampah organik, anorganik, dan B3 (bahan berbahaya dan beracun).



Gambar 1. Tahapan Pengumpulan dan Persiapan *Dataset*

2.1.1 Mengumpulkan Sampel Sampah

Penelitian ini menggunakan 50 sampel sampah dengan objek-objek sampah yang spesifik pada masing-masing sampel. Adapun rincian sampel sampah yang digunakan adalah sebagai berikut:

- Sampah organik memiliki 20 sampel sampah; cangkang pala, daun kering, daun segar, kantung teh, kayu, kulit alpukat, kulit bawang putih, kulit buah cokelat, kulit buah naga, kulit kacang, kulit lemon, kulit nanas, kulit pisang, kulit salak, kulit semangka, kulit telur, sabut kelapa, tempurung kelapa, tongkol jagung, tulang ikan.
- Sampah anorganik memiliki 20 sampel sampah; botol, bubblewrap, busa, gelas plastik, garpu, kantung plastik, kaleng, kardus, kertas, kemasan plastik, mika plastik, pipa, pulpen, sandal, sepatu, sendok, sisir, styrofoam, thinwall, seng.
- Sampah B3 memiliki 10 sampel sampah; aerosol, baterai, botol infus, kemasan salep, lampu, masker, obat-obatan strip, ponsel, suntik, termometer.

2.1.2 Pengambilan Gambar

Selanjutnya dilakukan pengambilan gambar terhadap setiap variasi objek pada masing-masing sampel sampah yang dikumpulkan menggunakan skenario pengambilan gambar dengan rincian pada Tabel 1.

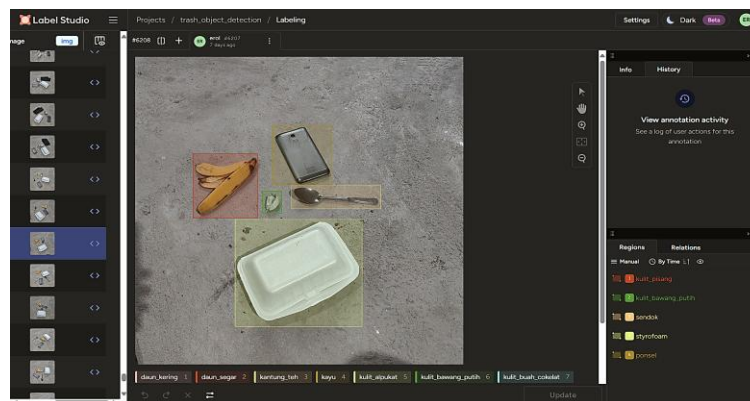
Tabel 1. Skenario Pengambilan Gambar

Skenario	Keterangan
Variasi objek sekaligus orientasi objek	Variasi dari 1 sampai 5 objek sampah yang terdiri dari objek sampah per sampel, gabungan dari objek sampah sejenis, dan gabungan dari semua jenis objek sampah dengan orientasi objek yang dimungkinkan.
Bidang latar belakang	Lantai cor beton dan lantai keramik warna putih.
Jarak pengambilan gambar	30 cm sampai 100 cm (tergantung jumlah dan ukuran objek)
Kondisi pencahayaan gambar	Terang

Proses pengumpulan gambar variasi objek pada masing-masing sampel sampah berdasarkan skenario pengambilan gambar pada Tabel 1, dilakukan menggunakan kamera perangkat Xiaomi Redmi Note 10 Pro.

2.1.3 Pelabelan dan Anotasi

Data gambar-gambar objek sampah yang terkumpul selanjutnya dipersiapkan dengan melakukan proses pelabelan dan anotasi menggunakan platform Label Studio.



Gambar 2. Proses Pelabelan dan Anotasi *Dataset*

Gambar 2 merupakan proses pelabelan dan anotasi *dataset* menggunakan platform Label Studio, terhadap semua objek dalam setiap gambar pada *dataset*. Proses ini terdiri dari melabel atau menamai setiap objek dan menerapkan koordinat objek atau *bounding box*. Setelah melakukan proses pelabelan dan anotasi, *dataset* kemudian di *export* menggunakan opsi *dataset* Pascal VOC.

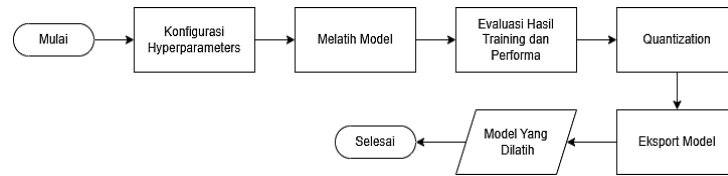
2.1.3 Pembagian Dataset

Tahapan terakhir dari pengumpulan dan persiapan *dataset* adalah melakukan pembagian data ke dalam dua set data yaitu *training set* dan *validation set*. Penelitian ini menggunakan rasio pembagian data 80:20, 80 untuk *training set* dan 20 untuk *validation set*.

2.2 Pelatihan Algoritma SSD MobileNetV2

SSD MobileNetV2 merupakan algoritma *object detection* gabungan dari arsitektur *convolutional neural network* MobileNetV2 yang bertugas untuk mengekstrak fitur dari gambar dan *single shot detector* (SSD) yang bertugas untuk melakukan lokalisasi dan klasifikasi terhadap fitur-fitur yang diekstrak oleh MobileNetV2 (Sandler et al., 2018). Pelatihan algoritma atau model SSD MobileNetV2 dilakukan menggunakan platform Kaggle dengan memanfaatkan

accelerator T4 GPU. Penelitian ini menggunakan *framework* MediaPipe dalam proses pelatihan model. MediaPipe merupakan sebuah *framework* yang dirancang untuk menyederhanakan pembangunan aplikasi yang bergantung pada pemrosesan data sensorik. MediaPipe memberikan fasilitas untuk pemrosesan data audio dan video secara *real-time*, yang memungkinkan pengembang untuk membangun aplikasi yang kompleks secara efisien di berbagai *platform* seperti lingkungan perangkat *mobile* (Lugaresi et al., 2019). *Framework* MediaPipe menyediakan *library* pelatihan model yaitu *mediapipe-model-maker* yang digunakan untuk melatih model dalam penelitian ini.



Gambar 3. Tahapan Pelatihan Algoritma SSD MobileNetV2

2.2.1 Konfigurasi *Hyperparameters*

Tahapan pertama dari pelatihan algoritma atau model SSD MobileNetV2 dimulai dengan melakukan konfigurasi *hyperparameters*.

Tabel 2. Konfigurasi *Hyperparameters*

<i>Hyperparameters</i>	Nilai
BATCH_SIZE	16
LEARNING_RATE	0.1
EPOCH	80

Berdasarkan Tabel 2 penelitian ini menggunakan nilai 16 untuk *hyperparameters* BATCH_SIZE yang berfungsi untuk membagi data *training set* menjadi kelompok data yang berisi 16 data per kelompok. Nilai 0,1 untuk *hyperparameters* LEARNING_RATE berfungsi untuk menyesuaikan nilai parameter *weight* dan *bias* dalam algoritma dengan nilai 0,1 pada proses *backpropagation*. Nilai 80 untuk *hyperparameters* EPOCH berfungsi untuk melakukan iterasi pelatihan model sebanyak 80 kali. Penelitian ini menggunakan nilai statis pada *hyperparameters* karena keterbatasan *resource platform* Kaggle yang digunakan untuk pelatihan model.

2.2.1 Melatih Model

Tahapan selanjutnya adalah melatih model, *library* *mediapipe-model-maker* menyediakan 2 varian model MOBILENET_V2, yang pertama MOBILENET_V2 yang memiliki input 256x256 dan kedua MOBILENET_V2_I320 yang memiliki input 320x320 (Google, 2025), penelitian ini menggunakan opsi model MOBILENET_V2_I320.

2.2.2 Evaluasi Hasil *Training Model dan Performa*

Berikutnya mengevaluasi hasil *training* dengan menggunakan *tools* bawaan *library* *mediapipe-model-maker*. Pada tahapan ini dievaluasi hasil *history training* model untuk mengetahui kemampuan model untuk belajar atau dilatih menggunakan *dataset* yang dibuat, juga mengevaluasi performa dari model yang telah dilatih.

2.2.3 *Quantization*

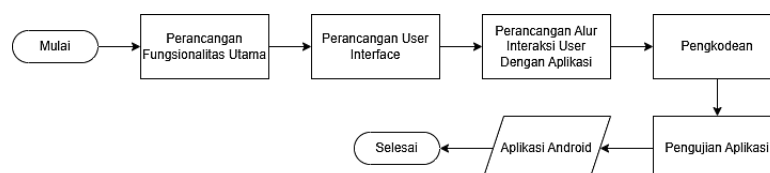
Tahapan selanjutnya dari proses pelatihan algoritma atau model adalah melakukan *quantization*. Tujuan dilakukannya proses *quantization* adalah untuk mereduksi ukuran model dan meningkatkan kecepatan prediksi dari model, *library* *mediapipe-model-maker* menyediakan dua bentuk teknik *quantization* yaitu *quantization aware training* yang optimal untuk penggunaan dengan CPU dan *post-training quantization* yang optimal untuk penggunaan dengan GPU (Google, 2025), penelitian ini menggunakan *post-training quantization*.

2.2.3 Eksport Model

Tahapan terakhir dari proses pelatihan algoritma atau model adalah melakukan eksport model yang sudah dilatih dan dikuantisasi, tahapan ini menghasilkan model dengan format *tflite*.

2.3 Pengembangan Aplikasi Android

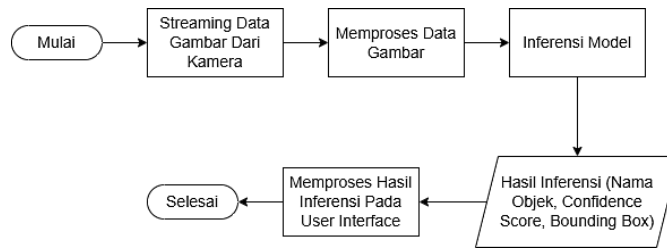
Berikut Gambar 4 merupakan tahapan pengembangan.



Gambar 4. Tahapan Pengembangan Aplikasi Android

2.3.1 Perancangan Fungsionalitas Utama

Tahapan pertama dari pengembangan aplikasi android dimulai dengan merancang alur interaksi fitur atau fungsionalitas utama yaitu proses *real-time* deteksi objek sampah.



Gambar 5. Proses *Real-Time* Deteksi

Gambar 5 merupakan alur proses *real-time* deteksi objek sampah aplikasi ini dimana aplikasi melakukan *streaming* data gambar dari kamera perangkat, selanjutnya setiap data *frame* gambar tersebut diproses untuk disesuaikan dengan kondisi data gambar yang bisa diterima model, kemudian model melakukan inferensi atau memproses setiap *frame* gambar tersebut dan menghasilkan hasil inferensi yang berisi nama objek, *confidence score*, dan koordinat objek atau *bounding box*, kemudian hasil inferensi ini dikelola pada *user interface*.

Selanjutnya melakukan konfigurasi fitur *real-time* deteksi, penelitian ini menggunakan *library* Mediapipe Tasks Vision untuk integrasi model yang sudah dilatih dengan aplikasi android, pada *library* tersebut digunakan kelas *ObjectDetector* untuk melakukan *real-time* deteksi objek.

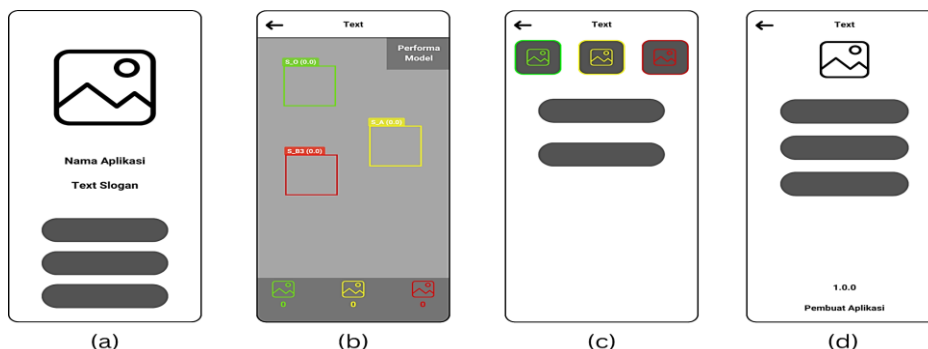
Tabel 3. Konfigurasi Kelas *ObjectDetector*

Parameter	Nilai
<i>runningMode</i>	LIVE_STREAM
<i>threshold</i>	0.6
<i>maxResults</i>	10
<i>currentDelegate</i>	GPU

Berdasarkan Tabel 3 nilai LIVE_STREAM pada parameter *runningMode* berfungsi untuk memasukan data *frame* gambar secara terus menerus ke model. Nilai 0,6 pada parameter *threshold* berfungsi untuk menampilkan objek yang dideteksi jika *confidence score* objek tersebut lebih dari 0,6 atau 60%. Nilai 10 pada parameter *maxResults* berfungsi untuk menampilkan maksimal 10 objek yang dideteksi. Nilai GPU pada parameter *currentDelegate* berfungsi untuk menggunakan GPU perangkat untuk melakukan proses inferensi model.

2.3.2 Perancangan *User Interface*

Tahapan selanjutnya merancang *user interface* aplikasi android, perancangan *user interface* adalah proses mendesain tampilan visual aplikasi yang berinteraksi langsung dengan pengguna. Perancangan *user interface* dilakukan menggunakan *wireframe*, *wireframe* berperan sebagai *blueprints* visual yang krusial dalam menyampaikan hierarki, struktur, dan fungsionalitas UI (Almani & Alrwais, 2024).

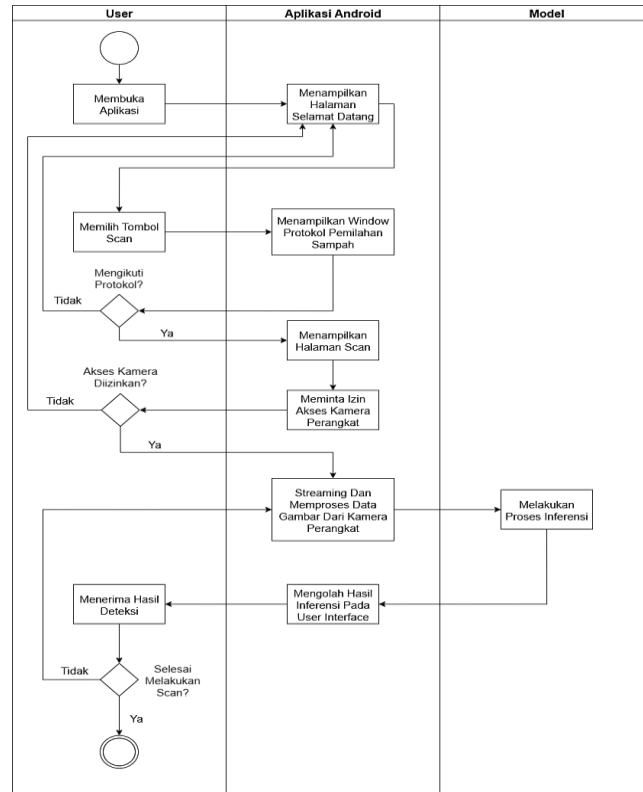


Gambar 6. *Wireframe User Interface* Aplikasi

Gambar 6 merupakan *wireframe user interface* aplikasi yang memiliki 4 halaman, terdiri dari (a) halaman selamat datang, (b) halaman *scan* atau *real-time* deteksi objek, (c) halaman edukasi pemilahan sampah, dan (d) halaman tentang aplikasi.

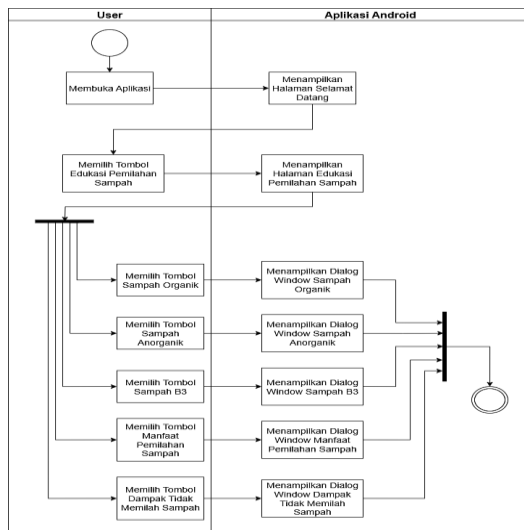
2.3.3 Perancangan Alur Interaksi User Dengan Aplikasi

Tahapan selanjutnya adalah merancang alur interaksi antara *user* dengan aplikasi menggunakan *activity diagram* untuk fungsionalitas utama dan fitur lainnya.

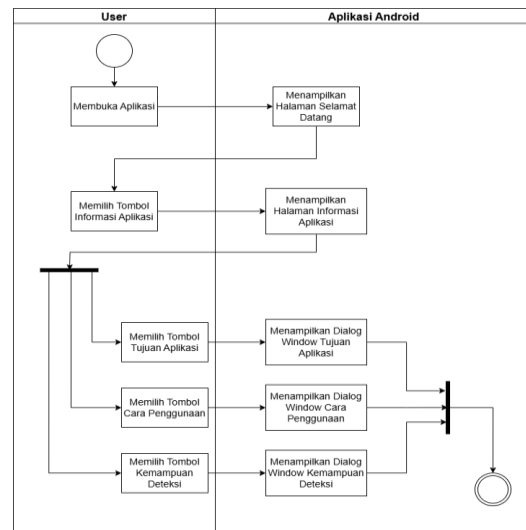


Gambar 7. Activity Diagram Halaman Scan

Gambar 7 merupakan diagram aktifitas yang bisa dilakukan *user* dalam menggunakan fitur *scan*, yang dimulai dengan *user* membuka aplikasi, kemudian memilih tombol *scan*, selanjutnya aplikasi akan meminta izin akses kamera perangkat, jika izin diberikan aplikasi akan memulai proses *scan* atau *real-time* deteksi objek sampah, kemudian *user* bisa menyudahi proses *scan*.



Gambar 8. Activity Diagram Halaman Edukasi Pemilahan Sampah



Gambar 9. Activity Diagram Halaman Tentang Aplikasi

Gambar 8 merupakan diagram aktifitas yang bisa dilakukan *user* dalam mengakses halaman edukasi pemilahan sampah. Gambar 9 merupakan diagram aktifitas yang bisa dilakukan *user* dalam mengakses halaman tentang aplikasi.

2.3.4 Pengkodean

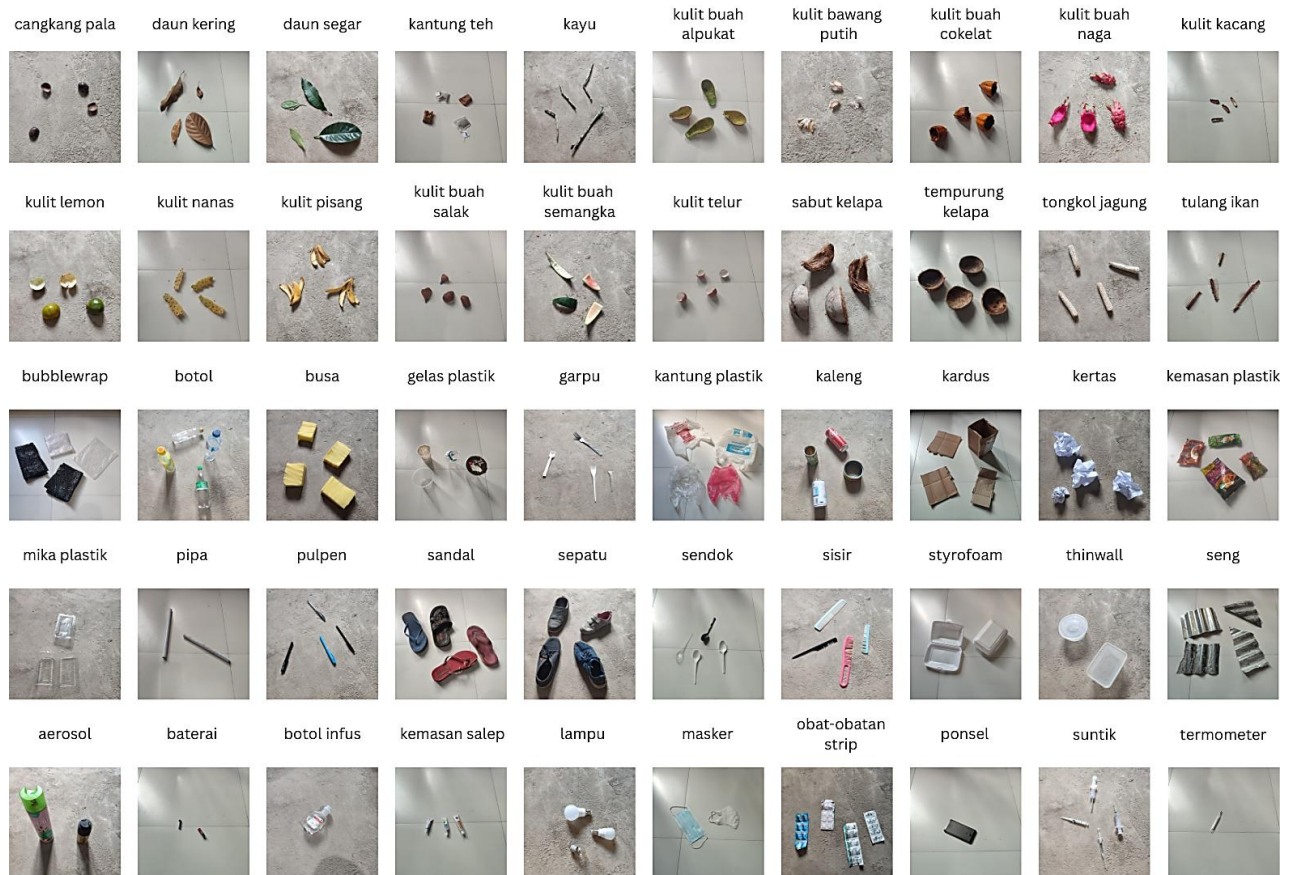
Tahapan selanjutnya adalah melakukan pengkodean atau mengimplementasikan rancangan-rancangan yang dibuat. Penelitian ini menggunakan *platform* Android Studio dan Visual Studio Code. Aplikasi ini dikembangkan menggunakan bahasa pemrograman Kotlin, *libraries* yang digunakan terdiri dari Jetpack Compose untuk *user interface*, CameraX untuk pemanfaatan kamera perangkat, dan MediaPipe Tasks Vision untuk inferensi model.

2.3.5 Pengujian Aplikasi

Tahapan terakhir dari pengembangan aplikasi android adalah melakukan pengujian, penelitian ini menggunakan pengujian yang terdiri dari *blackbox testing* untuk menguji fitur aplikasi, *performance testing* untuk menguji performa model dalam aplikasi, dan *detection testing* untuk menguji kemampuan deteksi objek aplikasi.

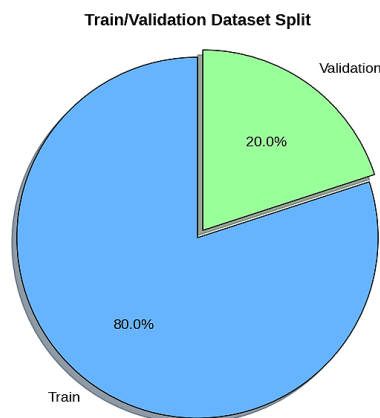
3. HASIL DAN PEMBAHASAN

3.1 Dataset



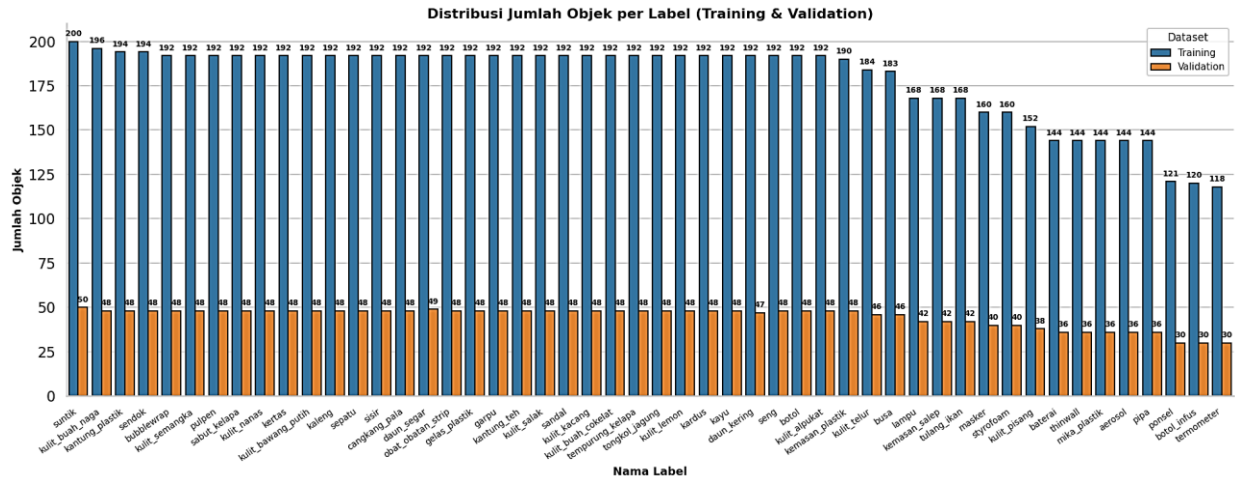
Gambar 10. Visualisasi *Dataset*

Gambar 10 merupakan visualisasi *dataset* yang dihasilkan berdasarkan skenario pengambilan gambar yang digunakan. *Dataset* dengan format Pascal VOC yang dibuat memiliki jumlah 4302 gambar objek-objek sampah, *dataset* ini memiliki 50 kelas atau label yang sesuai dengan sampel objek-objek sampah yang ditetapkan.



Gambar 11. Diagram Pembagian *Dataset*

Berdasarkan Gambar 11 *dataset* dibagi ke dalam *training set* dan *validation set* dengan alokasi data untuk *training set* sebanyak 3442 data atau 80% dan untuk *validation set* sebanyak 860 atau 20%.

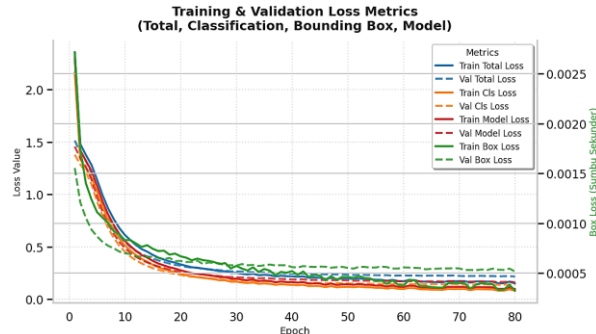


Gambar 12. Distribusi Objek *Dataset*

Gambar 12 merupakan diagram distribusi jumlah objek setiap label dalam *training set* dan *validation set*. Berdasarkan diagram tersebut dapat diamati terjadi ketidakseimbangan jumlah objek sampah pada beberapa label, ketidakseimbangan ini terjadi karena pada tahapan pengambilan gambar ada beberapa label sampah yang hanya memiliki sedikit variasi objek. Namun, *dataset* yang dibuat dapat digunakan untuk melatih model deteksi objek.

3.2 Model

Dataset yang telah dibuat kemudian dilatihkan pada algoritma atau model SSD MobileNetV2, menggunakan *library* mediapipe-model-maker. Model yang dilatih menggunakan *dataset* yang dibuat, konfigurasi *hyperparameters* yang digunakan, dan *accelerator* T4 GPU *platform* Kaggle, berlangsung selama 1 jam 46 menit 42 detik. Selanjutnya dilakukan evaluasi hasil *training* model menggunakan *tools* evaluasi bawaan mediapipe-model-maker yang terdiri dari *metrics total loss, classification loss, bounding box loss, dan model loss*.



Gambar 13. *History Training Model*

Berdasarkan Gambar 13 pada *epoch* 1 nilai *total loss* untuk *training set* 2,35 dan *validation set* 1,51, pada *epoch* terakhir nilai untuk *training set* 0,15 dan *validation set* 0,21. Nilai *classification loss* pada *epoch* 1 untuk *training set* 2,16 dan *validation set* 1,37, pada *epoch* terakhir nilai untuk *training set* 0,08 dan *validation set* 0,13. Nilai *bounding box loss* pada *epoch* 1 untuk *training set* 0,002 dan *validation set* 0,0015, pada *epoch* terakhir nilai untuk *training set* 0,0003 dan *validation set* 0,0005. Nilai *model loss* pada *epoch* 1 untuk *training set* 2,29 dan *validation set* 1,45, pada *epoch* terakhir nilai untuk *training set* 0,09 dan *validation set* 0,16. Berdasarkan hasil *history training* model menunjukkan algoritma SSD MobileNetV2 berhasil belajar atau dilatih menggunakan *dataset* yang dibuat.

```

Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.847
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=100 ] = 0.986
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=100 ] = 0.969
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.684
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.852
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.602
    
```

Gambar 14. Performa Model

Hasil evaluasi performa model pada Gambar 11 menunjukkan model memiliki kemampuan deteksi objek yang baik terutama pada metrik AP IoU=0.50:0.95 dengan nilai 0.847 yang menunjukkan hasil deteksi objek yang baik pada semua tingkat ambang batas, AP IoU=0.50 dengan nilai 0,986 yang menunjukkan hasil deteksi objek yang baik pada tingkat ambang batas IoU 50%, dan AP IoU=0.75 dengan nilai 0,969 yang menunjukkan hasil deteksi objek yang baik

pada tingkat ambang batas IoU 75%. Dengan demikian algoritma SSD MobileNetV2 yang telah dilatih bisa digunakan dan diintegrasikan dengan aplikasi Android.

3.3 Aplikasi Android

Algoritma atau model SSD MobileNetV2 yang telah dilatih menggunakan *dataset* yang dibuat kemudian diintegrasikan dengan aplikasi Android. Aplikasi Android yang dikembangkan diberi nama AI Pilah Sampah, aplikasi ini memiliki 4 halaman sesuai dengan *wireframe user interface*. Fungsionalitas utama aplikasi ini adalah melakukan *real-time* deteksi objek sampah dengan memanfaatkan algoritma SSD MobileNetV2 yang diintegrasikan. Mekanisme klasifikasi jenis sampah aplikasi ini adalah dengan menggunakan warna *bounding box* sesuai dengan jenis sampah objek yang dideteksi, terdiri dari warna hijau untuk sampah organik, warna kuning untuk sampah anorganik, dan warna merah untuk sampah B3, beserta label atau nama objek dan *confidence score* dengan mengikuti konfigurasi *real-time* deteksi library Mediapipe Tasks Vision yang dibuat. Selain fungsionalitas utama tersebut aplikasi ini juga memiliki halaman edukasi pemilahan sampah dan halaman tentang aplikasi.



Gambar 15. Halaman Selamat Datang



Gambar 16. Halaman Scan



Gambar 17. Halaman Edukasi Pemilahan Sampah



Gambar 18. Halaman Tentang Aplikasi

Gambar 12 merupakan halaman selamat datang yang ditampilkan ketika *user* membuka aplikasi ini. Gambar 13 merupakan halaman *scan* atau *real-time* deteksi objek yang merupakan fitur atau fungsionalitas utama aplikasi ini, ketika *user* membuka halaman ini algoritma SSD MobileNetV2 akan aktif dan melakukan pemrosesan deteksi objek sampah. Gambar 14 merupakan halaman edukasi pemilahan sampah yang berisi informasi mengenai pemilahan sampah. Gambar 15 merupakan halaman tentang aplikasi yang berisi informasi tentang aplikasi ini.

3.4 Pengujian Aplikasi Android

Tabel 4. Blackbox Testing

No.	Fitur	Hasil Yang Diharapkan	Status
1.	Navigasi ke halaman deteksi, halaman edukasi pemilahan sampah, dan halaman informasi aplikasi	Aplikasi berhasil membuka setiap halaman yang dipilih	Berhasil
2.	Perizinan akses kamera (izin diberikan)	Aplikasi melakukan proses <i>scan</i>	Berhasil
3.	Perizinan akses kamera (izin ditolak)	Aplikasi mengalihkan <i>user</i> ke halaman selamat datang	Berhasil
4.	<i>Preview</i> kamera	Aplikasi menampilkan tampilan kamera secara <i>real-time</i>	Berhasil
5.	Jumlah objek sampah organik, anorganik, dan B3 yang dideteksi	Aplikasi menampilkan jumlah objek sampah per jenis yang dideteksi secara <i>real-time</i>	Berhasil
6.	Performa model (fps, <i>inference time</i> , <i>detections</i>)	Aplikasi menampilkan fps, waktu inferensi model, dan jumlah semua objek yang dideteksi secara <i>real-time</i>	Berhasil
7.	<i>Tooltips icons</i> sampah organik, anorganik, dan B3	Aplikasi menampilkan daftar label sampah pada setiap jenis sampah	Berhasil
8.	Warna <i>overlay bounding box</i> , organik (hijau), anorganik (kuning), B3 (merah)	Aplikasi dapat mengklasifikasi jenis sampah dengan warna <i>bounding box</i> sesuai jenis objek sampah yang dideteksi secara <i>real-time</i>	Berhasil
9.	Maksimum deteksi 10 objek	Aplikasi dapat mendeteksi 10 objek sampah	Berhasil

Berdasarkan hasil pengujian *blackbox testing* pada Tabel 4 menunjukkan bahwa setiap fitur yang diuji berjalan sesuai yang diharapkan hal ini menunjukkan aplikasi telah memenuhi kebutuhan fungsional yang telah ditentukan, terutama pada fitur atau fungsionalitas utama aplikasi.

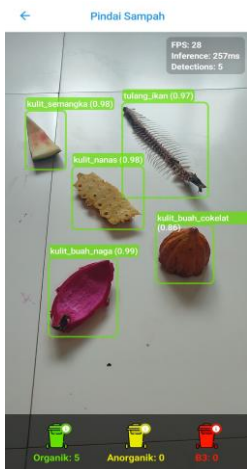
Berikutnya dilakukan pengujian kinerja algoritma atau model SSD MobileNetV2 dalam aplikasi, menggunakan *performance testing*. Pengujian ini dilakukan pada tiga perangkat ponsel Android dengan menjalankan fitur *scan*, ketiga perangkat yang digunakan memiliki spesifikasi yang berbeda dalam pengujian kinerja model untuk mendeteksi objek sampah secara *real-time*. Parameter yang diukur meliputi jumlah objek, FPS atau *frame per second*, dan *inference time*.

Tabel 5. Performance Testing

Perangkat Android	Jumlah Objek	FPS	Inference Time
Xiaomi Redmi Note 10 Pro	3	29.52	201.71 ms
Xiaomi Redmi Note 10 Pro	5	30.24	225.8 ms
Xiaomi Redmi Note 10 Pro	7	29.77	229.9 ms
Xiaomi Redmi Note 14 Pro+ 5G	3	30.74	166.2 ms
Xiaomi Redmi Note 14 Pro+ 5G	5	30.4	165.37 ms
Xiaomi Redmi Note 14 Pro+ 5G	7	30.66	175 ms
Xiaomi Redmi 13C	3	29.72	324.62 ms
Xiaomi Redmi 13C	5	29.84	331.38 ms
Xiaomi Redmi 13C	7	29.95	375.78 ms

Berdasarkan pengujian *performance testing* pada Tabel 5 dapat diamati bahwa nilai FPS rata-rata pada ketiga perangkat android yang digunakan relatif stabil berada di kisaran 29-31 FPS, hal ini menunjukkan bahwa aplikasi mampu menjalankan fitur *scan* atau *real-time* deteksi objek dengan baik. Namun, terjadi peningkatan waktu inferensi pada setiap perangkat untuk setiap jumlah objek yang dideteksi, hal ini menunjukkan peningkatan beban komputasi yang harus diproses oleh model SSD MobileNetV2. Terdapat juga perbedaan yang cukup signifikan pada waktu inferensi antar perangkat, perangkat Xiaomi Redmi Note 14 Pro+ 5G menunjukkan performa yang paling baik dengan waktu inferensi rata-rata terendah, yaitu sekitar 165-175 ms, yang menunjukkan perangkat ini memiliki proses komputasi model dengan efisiensi tertinggi di antara ketiga perangkat yang diuji. Sementara itu, perangkat Xiaomi Redmi Note 10 Pro memiliki waktu inferensi sedikit lebih tinggi, yakni berkisar antara 201-230 ms, namun masih tergolong dalam kategori yang dapat diterima untuk aplikasi *real-time*. Perangkat Xiaomi Redmi 13C mencatat waktu inferensi tertinggi, yaitu dalam rentang 324-375 ms, yang menunjukkan keterbatasan kemampuan komputasi perangkat tersebut. Berdasarkan pengujian *performance testing* didapatkan bahwa, perbedaan perangkat keras terutama GPU perangkat Android sangat mempengaruhi waktu pemrosesan yang dilakukan model. Perangkat dengan kapabilitas komputasi yang lebih tinggi cenderung lebih baik dalam menjalankan aplikasi pendeteksi dan pengklasifikasi sampah ini. Perangkat Android dengan spesifikasi menengah ke atas menunjukkan kinerja yang lebih optimal dalam menjalankan aplikasi ini dibandingkan dengan perangkat Android yang memiliki spesifikasi lebih rendah, yang berdampak kepada pengalaman pengguna dalam memilah sampah.

Berikutnya dilakukan pengujian kemampuan deteksi objek terhadap algoritma SSD MobileNetV2 yang terintegrasi dalam aplikasi Android, menggunakan *detection testing* dengan cara menjalankan fitur *scan* terhadap 5 objek sampah dengan menggunakan 1 objek sampah di setiap label menggunakan perangkat Xiaomi Redmi Note 10 Pro.



Gambar 19. Detection Testing 1



Gambar 20. Detection Testing 2



Gambar 21. Detection Testing 3

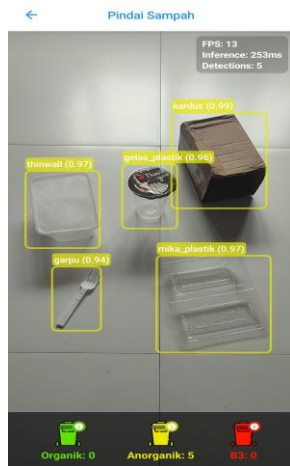


Gambar 22. Detection Testing 4

Hasil *detection testing* pada Gambar 19 menunjukkan aplikasi dapat mendeteksi objek-objek sampah beserta *confidence score* yang terdiri dari: kulit_semangka (0.98), tulang_ikan (0.97), kulit_nanas (0.98), kulit_buah_cokelat

(0.86), kulit_buah_naga (0.99). Hasil *detection testing* pada Gambar 20 menunjukkan aplikasi dapat mendeteksi objek-objek sampah beserta *confidence score* yang terdiri dari: kayu (0.94), tempurung_kelapa (0.93), kulit_alpukat (0.83), kulit_bawang_putih (0.97), cangkang_pala (0.88). Hasil *detection testing* pada Gambar 21 menunjukkan aplikasi dapat mendeteksi objek-objek sampah beserta *confidence score* yang terdiri dari: daun_kering (0,95), kulit_salak (0,84), kantung_teh (0,96), kulit_lemon (0,91), dan kulit_kacang (0,97). Hasil *detection testing* pada Gambar 22 menunjukkan aplikasi dapat mendeteksi objek-objek sampah beserta *confidence score* yang terdiri dari: daun_segar (0,90), sabut_kelapa (0,87), tongkol_jagung (0,88), kulit_pisang (0,99), dan kulit_telur (0,77).

Berdasarkan hasil *detection testing* pada ke empat gambar, menunjukkan aplikasi berhasil mendeteksi objek-objek sampah berjenis organik dengan benar dan *confidence score* yang tinggi.



Gambar 23. Detection Testing 5



Gambar 24. Detection Testing 6



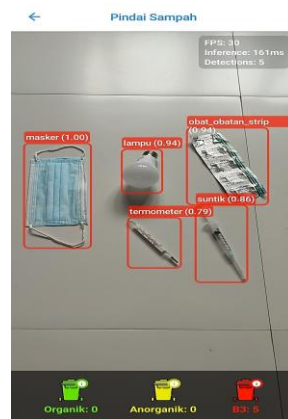
Gambar 25. Detection Testing 7



Gambar 26. Detection Testing 8

Hasil *detection testing* pada Gambar 23 menunjukkan aplikasi dapat mendeteksi objek-objek sampah beserta *confidence score* yang terdiri dari: thinwall (0.97), gelas_plastik (0.96), kartus (0.99), garpu (0.94), mika_plastik (0.97). Hasil *detection testing* pada Gambar 24 menunjukkan aplikasi dapat mendeteksi objek-objek sampah beserta *confidence score* yang terdiri dari: seng (0.96), sepatu (0.77), sendok (0.95), kemasan_plastik (0.92), styrofoam (0.96). Hasil *detection testing* pada Gambar 25 menunjukkan aplikasi dapat mendeteksi objek-objek sampah beserta *confidence score* yang terdiri dari: botol (0,88), busa (0,83), bubblewrap (0,93), kertas (0,99), dan pulpen (0,96). Hasil *detection testing* pada Gambar 26 menunjukkan aplikasi dapat mendeteksi objek-objek sampah beserta *confidence score* yang terdiri dari: kaleng (0,93), kantong_plastik (0,87), sisir (0,91), pipa (0,84), dan sandal (0,99).

Berdasarkan hasil *detection testing* pada ke empat gambar, menunjukkan aplikasi berhasil mendeteksi objek-objek sampah berjenis anorganik dengan benar dan *confidence score* yang tinggi.



Gambar 27. Detection Testing 9



Gambar 28. Detection Testing 10

Hasil *detection testing* pada Gambar 27 menunjukkan aplikasi dapat mendeteksi objek-objek sampah beserta *confidence score* yang terdiri dari: masker (1,0), lampu (0,94), obat_obatan_strip (0,94), suntik (0,86), dan termometer (0,79). Hasil *detection testing* pada Gambar 28 menunjukkan aplikasi dapat mendeteksi objek-objek sampah beserta *confidence score* yang terdiri dari: aerosol (0,95), botol_infus (0,87), baterai (0,74), ponsel (0,99), dan kemasan_salep (0,64).



Berdasarkan hasil *detection testing* pada ke dua gambar, menunjukkan aplikasi berhasil mendeteksi objek-objek sampah berjenis B3 dengan benar dan *confidence score* yang tinggi. Hasil *detection testing* yang dilakukan menunjukkan bahwa aplikasi mampu mendeteksi ke-50 objek sampah dengan benar dan *confidence score* yang tinggi. Namun, dalam proses *detection testing* ada sebagian objek yang tidak bisa langsung dideteksi oleh aplikasi ataupun salah dideteksi, diperlukan penyesuaian dari posisi perangkat, jarak antara objek dengan perangkat, orientasi objek, dan posisi objek, untuk bisa dideteksi dengan benar oleh aplikasi.

3.5 Pembahasan

Berdasarkan hasil penelitian, menunjukkan aplikasi pendeteksi dan pengklasifikasi sampah berbasis Android menggunakan algoritma SSD MobileNetV2 dapat menjadi solusi yang efisien dan praktis dalam mendukung edukasi pemilahan sampah dibandingkan dengan pendekatan konvensional yang membutuhkan banyak waktu, tenaga, dan sumber daya. Algoritma SSD MobileNetV2 yang dilatih menggunakan *dataset* yang dibuat menunjukkan kinerja deteksi yang baik, diukur menggunakan metrik AP IoU=0.50:0.95 didapatkan nilai mencapai 0,847. Hasil ini menunjukkan kemampuan model yang baik dalam mendeteksi objek sampah. Hasil *performance testing* menunjukkan bahwa integrasi algoritma SSD MobileNetV2 pada aplikasi Android menghasilkan performa inferensi yang baik, khususnya pada perangkat Android kelas menengah ke atas dengan waktu pemrosesan berkisar antara 165–230 ms, hal ini memberikan pengalaman yang optimal bagi pengguna dalam aktivitas pemilahan sampah secara *real time*. Hasil *detection testing* juga menunjukan aplikasi ini mampu mendeteksi ke 50 objek sampah yang dilatihkan pada model, hal ini memberikan pengalaman edukatif yang interaktif bagi pengguna dalam memilah sampahnya. Meskipun hasil pengujian menunjukkan hasil yang baik, aplikasi ini masih memiliki dua tantangan utama yaitu kinerja yang kurang optimal pada perangkat Android berspesifikasi rendah dan dalam melakukan deteksi sampah diperlukan penyesuaian terhadap beberapa objek sampah baik dari sisi perangkat maupun objek sampah untuk bisa dideteksi dengan benar oleh aplikasi.

4. KESIMPULAN

Penelitian ini menghasilkan aplikasi pendeteksi dan pengklasifikasi sampah berbasis android menggunakan algoritma SSD MobileNetV2. Aplikasi ini dikembangkan sebagai solusi yang lebih efisien dan lebih praktis terhadap edukasi sekaligus penanganan pemilahan sampah dibandingkan pendekatan konvensional yang membutuhkan banyak waktu, tenaga, dan sumber daya. Penelitian ini dimulai dengan mengumpulkan dan mempersiapkan *dataset* yang di-*export* menjadi format Pascal VOC dengan jumlah 4302 gambar objek-objek sampah, *dataset* kemudian dibagi menjadi 2 set data yaitu *training set* dengan alokasi 3442 data dan *validation set* 860 data. *Dataset* yang dibuat kemudian dilatihkan pada algoritma SSD MobileNetV2 menggunakan *framework* MediaPipe *library* mediapipe-model-maker, berdasarkan hasil evaluasi *training* menunjukan model berhasil belajar atau berlatih menggunakan *dataset* yang dibuat dengan performa AP IoU=0.50:0.95 mencapai 0.847, performa ini menunjukan kemampuan deteksi objek yang baik dari model, dengan demikian model bisa diintegrasikan dengan aplikasi android. Aplikasi android yang dikembangkan diberi nama AI Pilah Sampah, berdasarkan hasil pengujian *blackbox testing* didapatkan seluruh fitur aplikasi dapat berjalan dengan baik. Hasil pengujian *performance testing* menunjukan aplikasi yang dikembangkan cenderung lebih optimal dijalankan pada perangkat menengah ke atas, dengan hasil *inference time* berada pada rentang 165-230 ms. Hasil pengujian *Detection testing* didapatkan aplikasi mampu mendeteksi ke 50 objek sampah walaupun diperlukan penyesuaian pada sebagian objek sampah. Aplikasi ini memiliki fungsionalitas utama melakukan *real-time* deteksi terhadap objek-objek sampah dan mengklasifikasi jenisnya menggunakan warna *bounding box*, dengan mekanisme ini memberikan pengalaman yang interaktif kepada pengguna dalam memilah sampahnya, sehingga diharapkan meningkatkan kesadaran akan pemilahan sampah. Saran untuk penelitian selanjutnya yaitu menambah variasi objek dan label sampah untuk menambah pengetahuan dari model dan aplikasi, melakukan *hyperparameters tuning* ataupun menggunakan algoritma yang berbeda untuk mengoptimalkan performa dari model, melakukan optimalisasi aplikasi khususnya untuk perangkat Android dengan spesifikasi rendah, dan melakukan penelitian tingkat efektivitas pendekatan yang dihasilkan dalam penelitian ini dengan pendekatan edukasi dan penanganan pemilahan sampah lainnya.

REFERENCES

- Almani, A., & Alrwais, O. (2024). The Role of Wireframes in Enhancing User Interface Design. *International Research Journal of Innovations in Engineering and Technology*, 08(12), 134–140. <https://doi.org/10.47001/irjiet/2024.812020>
- Daeng, M., Rompas, P. T. D., & Santa, K. (2025). Tracking Dan Monitoring Truk Pengangkut Sampah Di Dinas Lingkungan Hidup Kabupaten Minahasa. *JOINTER: Journal of Informatics Engineering*, 6(01), 1–13. <https://doi.org/10.53682/jointer.v6i01.306>
- Google. (2025). *Object detection model customization guide*. https://ai.google.dev/edge/mediapipe/solutions/customization/object_detector
- Humas BRIN. (2022). *Tantangan dan Harapan Pengelolaan Sampah di Indonesia*. Badan Riset Dan Inovasi Nasional. <https://brin.go.id/news/105094/tantangan-dan-harapan-pengelolaan-sampah-di-indonesia>
- Kementerian Lingkungan Hidup dan Kehutanan. (2024). *Sistem Informasi Pengelolaan Sampah Nasional*.



<https://sipsn.kemendagri.go.id/sipsn/>

- Khan, D., Waqas, M., Tahir, M., Islam, S. U., Amin, M., Ishtiaq, A., Jan, L., & Latif, J. (2023). Revolutionizing Real-Time Object Detection: YOLO and MobileNet SSD Integration. *Journal of Computing & Biomedical Informatics*, 6(1). <https://doi.org/10.56979/601/2023>
- Khomsyi, S. N., Ramme, M. Y., Pandiangan, G. F., Wiyono, B. N., & Putri, A. L. R. (2024). Sosialiasi Pemilahan Sampah di Desa Kedungrandu: Solusi Pengelolaan dan Pemanfaatan Sampah Rumah Tangga. *IJCOSIN: Indonesian Journal of Community Service and Innovation*, 4(2), 15–23. <https://doi.org/10.20895/ijcosin.v4i1.1549>
- Kunwar, S. (2023). MWaste: An app that uses deep learning to manage household waste. *Clean Technologies and Recycling*, 3(3), 119–133. <https://doi.org/10.3934/ctr.2023008>
- Laksono, P. W., Anisa, A., & Priyandari, Y. (2024). Deep learning implementation using convolutional neural network in inorganic packaging waste sorting. *Franklin Open*, 8(August), 100146. <https://doi.org/10.1016/j.fraope.2024.100146>
- Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M., & Grundmann, M. (2019). *MediaPipe: A Framework for Building Perception Pipelines*. <http://arxiv.org/abs/1906.08172>
- Martina Lorensa, Rorimpandey, G. C., & Santa, K. (2024). Perbandingan Algoritma Regresi Linear dengan Algoritma Backpropagation dalam Estimasi Timbulan Sampah di Sulawesi Utara. *The Indonesian Journal of Computer Science*, 13(5), 8459–8479. <https://doi.org/10.33022/ijcs.v13i5.4093>
- Meilani, W., & Harianti, R. (2024). Pengaruh Penyuluhan Pemilahan Sampah Melalui Media Poster Terhadap Peningkatan Pengetahuan Siswa Sdn Wanajaya Iii Karawang. *Al-Tamimi Kesmas: Jurnal Ilmu Kesehatan Masyarakat (Journal of Public Health Sciences)*, 13(1), 129–138. <https://doi.org/10.35328/kesmas.v13i1.2678>
- Rosmaini, & Birman, Y. (2022). Pencegahan Pencemaran Lingkungan Melalui Pemberian Edukasi Pemilahan Sampah Denganmedia Brosur Di Kecamatan Bungus Kota Padang. *Jurnal Abdimas Sainatika*, 5(1), 101–115. <https://jurnal.syedzasainatika.ac.id/index.php/abdimas/article/view/1333/944>
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>
- Single, S., Iranmanesh, S., & Raad, R. (2023). RealWaste: A Novel Real-Life Data Set for Landfill Waste Classification Using Deep Learning. *Information (Switzerland)*, 14(12). <https://doi.org/10.3390/info14120633>
- Statcounter. (2025). *Mobile & Ta Operating System Market Share Indonesia*. Statcounter Global Stats. https://gs.statcounter.com/os-market-share/mobile-ta/indonesia?utm_source=chatgpt.com
- Szeliski, R. (2018). *Computer Vision: Algorithms and Applications*, 2nd Edition. Springer, Cap. 1. <https://www.wiley.com/en-in/Machine+Vision+Algorithms+and+Applications%2C+2nd+Edition-p-9783527413652>
- Tian, X., Shi, L., Luo, Y., & Zhang, X. (2024). Garbage Classification Algorithm Based on Improved MobileNetV3. *IEEE Access*, 12(March), 44799–44807. <https://doi.org/10.1109/ACCESS.2024.3381533>
- Walangitan, A. S. N., & Rompas, P. T. D. (2023). Pengembangan Aplikasi E-Commerce Berbasis Website Untuk Pengelolaan Sampah Berkelanjutan: Studi Kasus Dinas Lingkungan Hidup Kota Tomohon Menggunakan Metode Extreme Programming (XP). *Journal Of Social Science Research*, 3(6), 7383–7392.
- Yuliawati, I. S., Azahra, R., Rohmalia, F., Ajeng, K., Septiandari, R., Putri, F. A., & Kusuma, R. M. (2024). Penyuluhan Pentingnya Pengolahan Sampah Organik dan Non-Organik pada MI Darussalam Karanglo 2. *Jurnal Pengabdian Sosial*, 1(10), 1648–1654. <https://doi.org/10.59837/rssf6c40>