



RANCANG BANGUN APLIKASI KATALOG DAN KLASIFIKASI SAMPAH RUMAH TANGGA BERBASIS ANDROID MENGGUNAKAN KOTLIN DAN MOBILENET

Ichsan Ardika Akbar¹, Endang Wahyu Pamungkas²

¹Universitas Muhammadiyah Surakarta, Jl. A. Yani, Mendungan, Pabelan, Kec. Kartasura, Kabupaten Sukoharjo, Indonesia

²Universitas Muhammadiyah Surakarta, Jl. A. Yani, Mendungan, Pabelan, Kec. Kartasura, Kabupaten Sukoharjo, Indonesia

 iksanardika@gmail.com

Abstrak. Peningkatan populasi dan pola konsumsi masyarakat modern berkontribusi langsung terhadap lonjakan volume sampah harian. Namun, kesadaran dari masyarakat akan pentingnya pengelolaan sampah yang berkelanjutan masih tergolong rendah. Untuk mengatasi hal itu, diperlukan solusi praktis yang dapat mendorong kebiasaan memilah sampah langsung dari tingkat rumah tangga. Pada tugas akhir ini, dirancang sebuah aplikasi *mobile* berbasis Android yang mendukung proses klasifikasi dan pencatatan jenis sampah secara otomatis. Aplikasi ini menggunakan teknologi *machine learning* dengan model *MobileNet*, yang mampu membedakan kategori sampah berdasarkan citra gambar yang diambil melalui kamera perangkat. Pengembangan sistem ini mengikuti model *Software Development Life Cycle* (SDLC) tipe *Waterfall*, yang diintegrasikan dengan perancangan model *machine learning* mulai dari analisis kebutuhan, desain sistem, *coding*, implementasi, hingga tahap pengujian dan perawatan. Model *machine learning* yang diterapkan menunjukkan kinerja yang cukup baik, dengan rata-rata akurasi sebesar 94% pada data uji, dan mempertahankan akurasi sebesar 84% dalam lingkungan nyata. Evaluasi terhadap antarmuka dan kemudahan penggunaan dilakukan menggunakan metode *System Usability Scale* (SUS), yang menunjukkan respon positif dari pengguna. Hasil ini menunjukkan bahwa aplikasi tidak hanya mampu mendeteksi dan mengelompokkan sampah secara akurat, tetapi juga dapat menjadi sarana edukatif yang efektif untuk meningkatkan kesadaran dan partisipasi masyarakat dalam pengelolaan sampah secara mandiri dan berkelanjutan di era digital.

Kata kunci: *pengelolaan sampah; aplikasi Android; kotlin; machine learning; image classification*



PENDAHULUAN

Berdasarkan Undang-Undang Republik Indonesia Nomor 18 Tahun 2008 tentang pengolahan sampah, Sampah adalah sisa kegiatan sehari-hari manusia dan/atau proses alam yang berbentuk padat. Pada tahun 2023 jumlah penduduk Indonesia mencapai 281 juta jiwa, menjadikannya negara dengan populasi terbesar keempat di dunia [1]. Angka ini diperkirakan akan terus meningkat seiring perkembangan zaman. Dengan populasi yang besar, tingkat konsumsi di Indonesia juga tinggi, yang berdampak pada meningkatnya produksi sampah dari hasil konsumsi tersebut. Adanya penambahan penduduk dan perubahan pola konsumsi masyarakat mengakibatkan bertambahnya volume, jenis, dan karakteristik sampah yang semakin beragam. Setiap tahun, produksi sampah semakin tinggi, berbanding lurus dengan meningkatnya kegiatan konsumsi dalam masyarakat.

Masalah terkait pengolahan sampah selalu menjadi perhatian yang belum terselesaikan. Sering kali, masyarakat merasa abai terkait sampah yang mereka hasilkan. Akibatnya, sampah semakin hari semakin menumpuk sehingga lingkungan menjadi tempat yang kurang nyaman. Menurut data dari Kementerian Lingkungan Hidup/Badan Pengendalian Lingkungan Hidup Republik Indonesia, pada tahun 2024 sebanyak 32,6 Juta ton sampah rumah tangga dihasilkan. Dari angka tersebut, sampah yang terkelola sebanyak 19,5 juta ton yang berarti sebanyak 13,1 juta ton sampah masih belum dapat dikelola dengan baik [2]

Menurut Menteri Lingkungan Hidup Hanif Faisol Nurofiq melalui konferensi pers di Kantor Kementerian Lingkungan Hidup di Jakarta, Presiden Indonesia menargetkan angka penanganan sampah mendekati 100% pada tahun 2029 [3]. Angka tersebut masih jauh, mengingat penanganan sampah di Indonesia pada tahun 2024 masih berkutat di nilai 59%. Pemerintah sendiri sudah mengupayakan banyak fasilitas untuk mengelola sampah rumah tangga yang ada di masyarakat. Beberapa fasilitasnya antara lain bank sampah, komposting, pusat daur ulang, dan penyediaan tempat pemrosesan akhir (TPA) di berbagai tempat supaya sampah dapat diolah lebih lanjut. Namun fasilitas tersebut masih belum bisa menunjang pengolahan sampah hingga titik maksimal.

Banyak cara untuk kita membantu pemerintah mencapai target tersebut. Salah satunya adalah dengan memilah sampah yang kita hasilkan sedini mungkin. Pemilahan sampah yang berasal dari ranah pemukiman/tempat tinggal dan daerah komersial dapat dibagi menjadi 3 golongan, yaitu sampah organik, sampah anorganik, dan sampah B3 [4]. Sampah organik merupakan sampah yang berasal dari makhluk hidup, dapat membusuk dan mudah terurai misalnya sisa makanan dan dedaunan. Sampah anorganik adalah

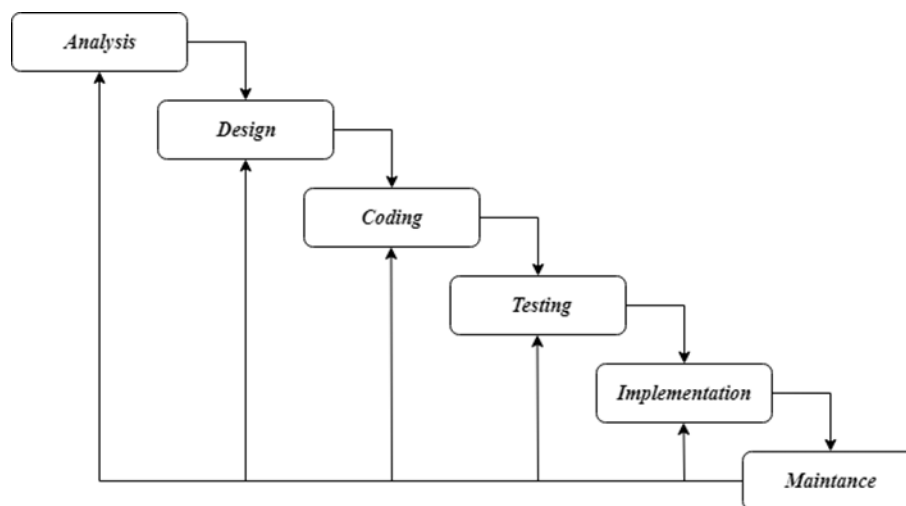


sampah yang berasal dari bahan-bahan non-hayati yang merupakan produk sintetis atau hasil dari pengolahan bahan tambang atau sumber daya alam yang susah terurai, contohnya kaca dan plastik. Sedangkan sampah B3 adalah singkatan dari sampah bahan berbahaya dan beracun, disebut demikian karena sampah ini berbahaya bagi lingkungan dan makhluk hidup termasuk manusia, karena sifat, konsentrasi, dan jumlahnya. Yang termasuk sampah B3 misalnya batu baterai, lampu bohlam, kemasan obat, dan sebagainya [5].

Tujuan dari tugas akhir ini adalah memberikan fasilitas untuk mempermudah pemilahan sampah dalam pengolahan sampah di tingkat terkecil dalam masyarakat yaitu perumahan dan rumah tangga. Sehingga sampah yang dihasilkan oleh rumah tangga lebih mudah diolah. Dengan adanya tugas akhir ini, diharapkan mampu memberikan kontribusi kepada negara dan masyarakat dalam mewujudkan lingkungan yang bersih dan bebas sampah.

METODE

Metode yang digunakan untuk perancangan dan pengembangan aplikasi klasifikasi sampah ini menggunakan prinsip *Software Development Life Cycle (SDLC) waterfall*. Model SDLC waterfall merupakan pendekatan pengembangan perangkat lunak yang bersifat sekuensial, di mana setiap tahap dilalui secara berurutan seperti aliran air terjun. Setiap fase dalam proses ini harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya, sehingga membentuk alur kerja yang sistematis untuk memastikan perangkat lunak dapat dibangun secara efektif dan terstruktur [6]. Pada gambar 1, disajikan ilustrasi mengenai alur kerja metode *waterfall*.



Gambar 1. Alur tahapan metode *waterfall* [7]



Menurut penelitian yang dilakukan oleh Samuli Laato (2022), perancangan model *machine learning* dapat diterapkan pada metode SDLC dengan menyesuaikan tahapan pengembangannya. Meskipun proses pengembangan model ML sering berjalan secara paralel dan tidak selalu mengikuti model SDLC secara eksplisit, alur kerjanya tetap tunduk pada struktur kerja formal tim pengembang. Kegiatan utama seperti pengumpulan data, rekayasa data, pelatihan, evaluasi model, hingga integrasi ke dalam sistem dapat diselaraskan dengan tahapan dalam metode SDLC, sehingga memungkinkan rekayasa model ML menjadi bagian dari pengembangan sistem aplikasi [8].

1. Analisis Kebutuhan

Pada tahap ini, peneliti secara sistematis mengidentifikasi dan menganalisis keseluruhan kebutuhan pengembangan, serta merumuskan secara rinci spesifikasi fungsional dan non-fungsional yang wajib dipenuhi [9]. Analisis dilakukan dengan melihat permasalahan mengenai sampah rumah tangga yang ada di lingkungan sekitar. Kemudian mengidentifikasi kebutuhan pengguna yang dapat diimplementasikan ke dalam sebuah sistem. Dalam konteks ini, kebutuhan sistem mencakup pengumpulan dan pelabelan data gambar yang akan digunakan sebagai dataset untuk training model klasifikasi. Selanjutnya, dataset yang telah dikumpulkan akan diproses dengan mekanisme pelatihan transfer learning yang dirancang untuk menghasilkan model klasifikasi yang ringan dan akurat, serta dapat diintegrasikan ke dalam platform *mobile*. Berikut adalah kebutuhan sistem yang diperlukan berdasarkan analisis yang telah dilakukan.

1.1 Kebutuhan Fungsional

- Sistem harus memungkinkan pengguna untuk mendaftar dan masuk ke aplikasi menggunakan email serta kata sandi.
- Sistem harus bisa mendapatkan akses ke internet.
- Sistem harus memiliki akses ke kamera dan galeri untuk mengambil atau memilih gambar.
- Sistem harus mampu melakukan klasifikasi gambar sampah secara otomatis kemudian menampilkan hasil klasifikasi dan rekomendasi sesuai jenis sampah
- Sistem harus memungkinkan pengguna untuk mengedit, menyimpan, menghapus, atau menambahkan data sampah mereka.

1.2 Kebutuhan Non-Fungsional

- Sistem harus dapat menjaga keamanan data pengguna.
- Sistem harus dapat memproses permintaan pengguna dengan cepat.



- Sistem harus dapat berjalan pada perangkat Android versi Android 7.0 atau yang lebih baru
- Sistem harus mempunyai tampilan yang intuitif, simpel, dan mudah digunakan
- Model klasifikasi harus memiliki tingkat akurasi yang tinggi
- Model yang digunakan harus disimpan dalam format TensorFlow Lite (.tflite) agar kompatibel dan efisien untuk digunakan pada perangkat Android.

2. Perancangan Desain

Tahap desain bertujuan untuk menciptakan sistem baru yang mampu mengatasi berbagai permasalahan yang mungkin dihadapi, sekaligus menentukan alternatif sistem yang paling optimal [10]. Dalam penelitian ini, proses perancangan mencakup dua aspek utama, yaitu perancangan sistem perangkat lunak aplikasi Android dan perancangan model klasifikasi berbasis *machine learning*.

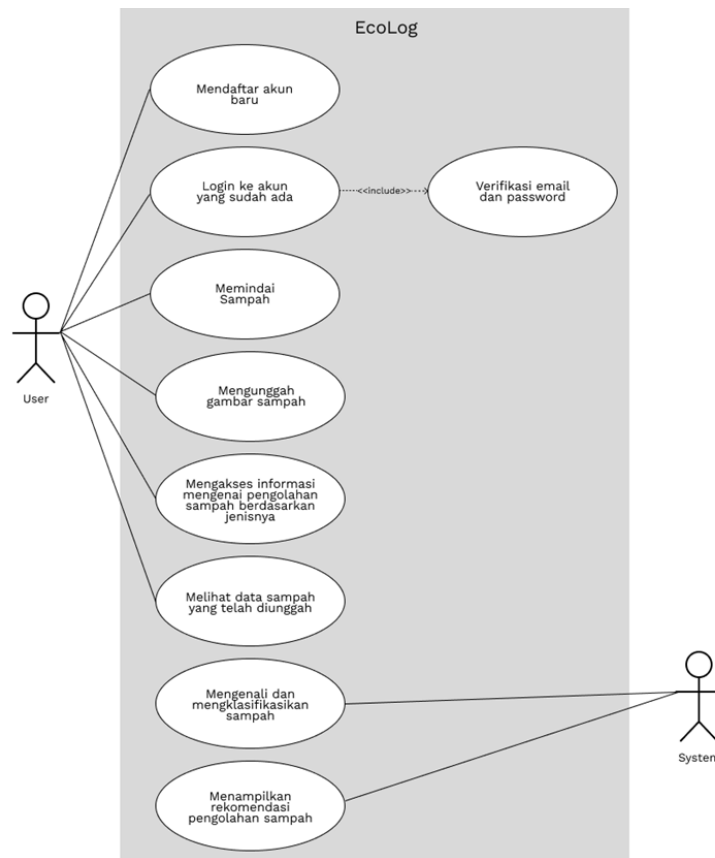
Perancangan model *machine learning* dalam aplikasi ini dilakukan melalui pendekatan *transfer learning* menggunakan arsitektur *pretrained convolutional neural network* (CNN), khususnya *MobileNetV2*. Proses perancangan model dimulai dari tahap pelabelan dan *preprocessing* data, di mana data sampah yang telah dikumpulkan dilabeli secara manual ke dalam 12 kelas yang merupakan sampah umum di lingkungan rumah tangga. Selanjutnya, melakukan *resize* terhadap data tersebut ke resolusi 224x224 piksel, normalisasi nilai piksel, dan *augmentasi menggunakan image data generator* untuk meningkatkan kemampuan generalisasi model.

Tahap selanjutnya adalah perancangan arsitektur model dengan memanfaatkan *backbone* dari *pretrained MobileNetV2* dan mengganti lapisan *output*-nya dengan dense layer yang jumlah neuronnya disesuaikan dengan jumlah kelas target. Fungsi aktivasi yang digunakan pada lapisan akhir adalah softmax guna mendukung klasifikasi multi-kelas. Model kemudian dilatih menggunakan dataset yang telah disediakan. Proses training menggunakan *optimizer* Adam dan fungsi *loss categorical cross-entropy*, sedangkan evaluasi performa dilakukan menggunakan *classification report*. Setelah model mencapai performa yang optimal, model dikonversi ke dalam format TensorFlow Lite (.tflite) agar dapat diintegrasikan ke dalam aplikasi *mobile*

Perancangan sistem aplikasi dilakukan menggunakan *use case diagram*, *activity diagram*, *entity relationship diagram* (ERD), dan *wireframe*. Setiap elemen desain menggambarkan alur interaksi pengguna dengan sistem serta pengelolaan data internal, termasuk data hasil klasifikasi. Perancangan sistem atau aplikasi dilampirkan pada poin poin dibawah ini.

2.1 Use Case Diagram





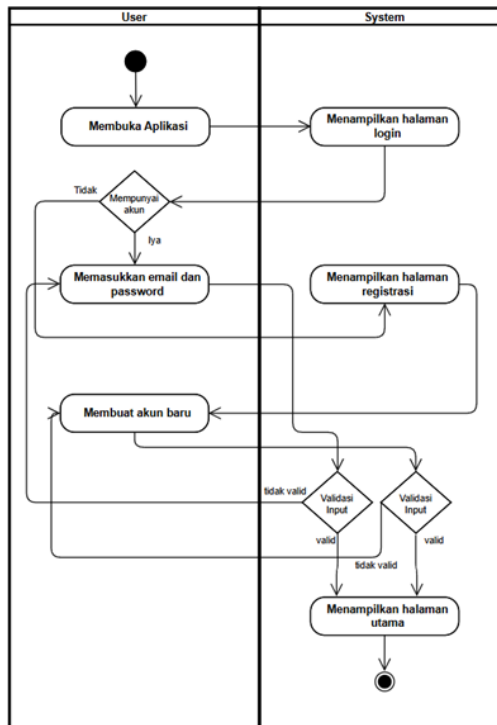
Gambar 2. Use case diagram

Use case diagram merupakan salah satu jenis diagram perilaku yang dibentuk berdasarkan analisis *use case*. Diagram ini bertujuan untuk memberikan representasi visual mengenai fungsi-fungsi yang dimiliki oleh suatu sistem, mencakup aktor yang terlibat, tujuan mereka, serta hubungan antar *use case* yang ada [11]. *Use case diagram* dari aplikasi rekomendasi pengolahan sampah EcoLog dilampirkan pada gambar 2

2.2 Activity Diagram

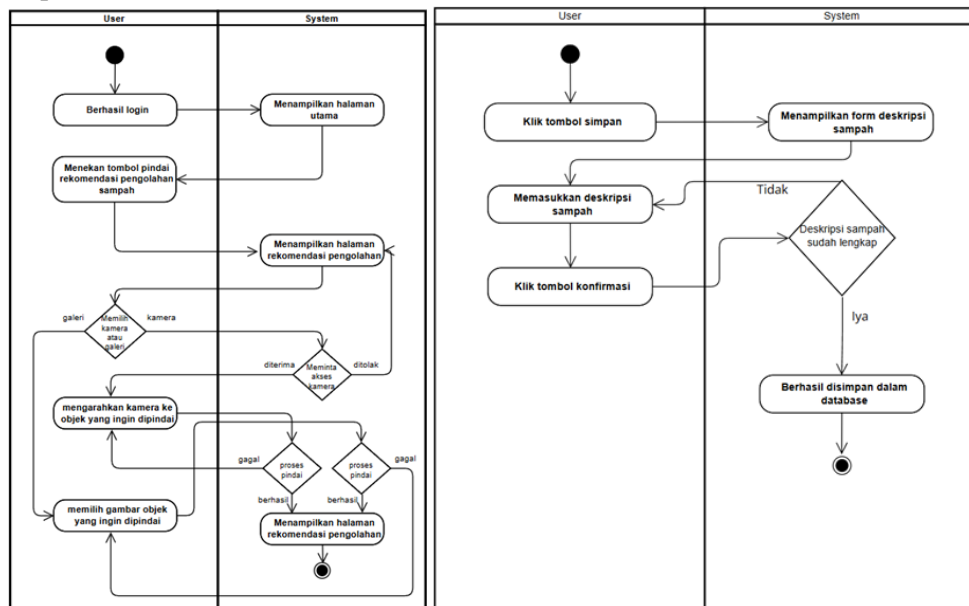
Activity diagram adalah jenis khusus dari *state diagram* yang digunakan untuk menunjukkan alur kerja atau proses. Diagram ini merupakan bagian dari pendekatan yang lebih luas untuk memodelkan perilaku sistem [12].





Gambar 3. Activity Diagram Login

Gambar 3 merupakan *activity diagram* untuk pengguna *login*. Pada *activity diagram* diatas, pengguna dapat melakukan *login* jika sudah memiliki akun, atau registrasi ketika belum memiliki akun. Setelah pengguna berhasil *login*, maka akan menampilkan halaman utama.



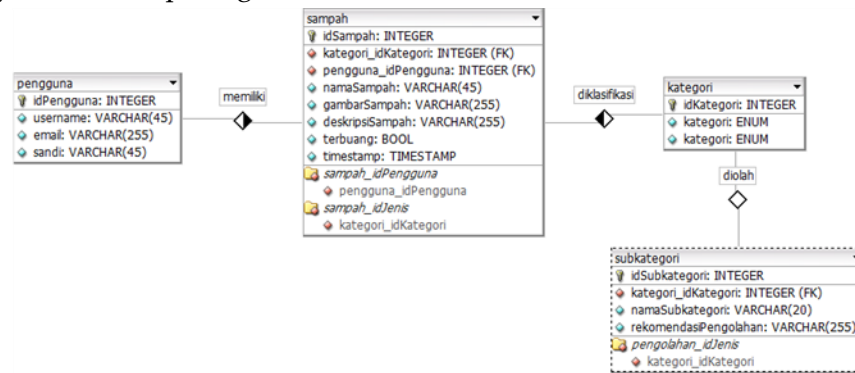
Gambar 4. Activity diagram klasifikasi dan simpan

Pada gambar 4, disajikan 2 buah *activity diagram* yaitu *activity diagram* klasifikasi sampah dan *activity diagram* klasifikasi menyimpan sampah. Setelah



pengguna berhasil *login*, akan langsung masuk ke halaman utama, dimana pengguna bisa menekan tombol pindai untuk mencoba melakukan klasifikasi sampah berdasarkan gambar, dan ketika proses pindai selesai akan menampilkan halaman rekomendasi pengolahan. Pada halaman rekomendasi pengolahan, terdapat tombol simpan. Setelah pengguna menekan tombol simpan, maka aplikasi akan menampilkan form untuk mengisi deskripsi dari sampah yang akan disimpan. Setelah pengguna mengisi deskripsi sampah dan menekan tombol konfirmasi, maka sampah berhasil disimpan.

2.3 Entity Relationship Diagram



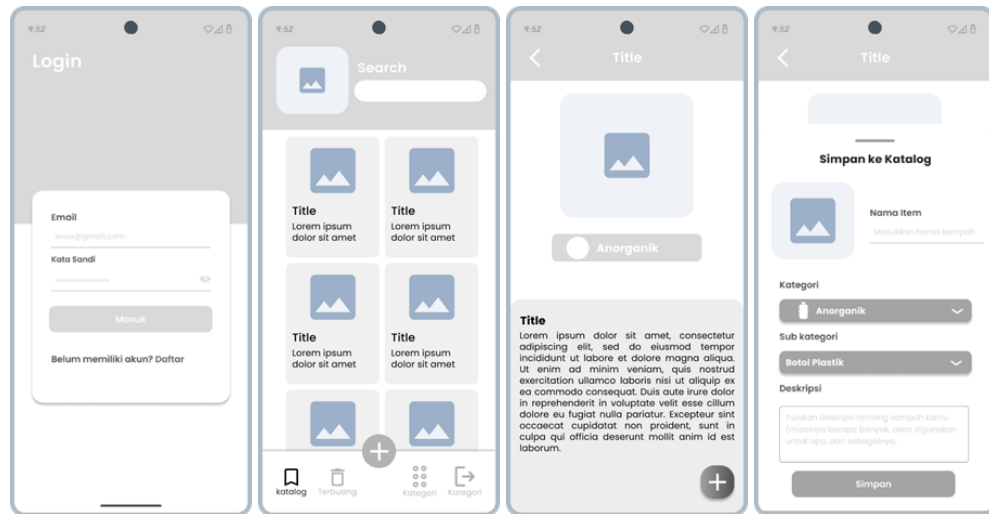
Gambar 5. Entity relationship diagram

Entity relationship diagram adalah model data konseptual tingkat tinggi yang menggambarkan data sebagai entitas, atribut, dan hubungan [13]. Pada gambar 5, disajikan ER diagram untuk aplikasi EcoLog. Pada ER diagram tersebut, terdapat 4 entitas yaitu pengguna, sampah, kategori, dan subkategori yang masing masing memiliki relasi yang berbeda beda.

2.4 Wireframe

Wireframe adalah representasi visual dasar dari tata letak *interface* sebuah situs web atau aplikasi [14]. Wireframe digunakan untuk mempermudah pembuatan *user interface* dan *user experience* lebih lanjut.





Gambar 6. Wireframe aplikasi

Pada gambar 6, *wireframe* aplikasi EcoLog terdiri dari 4 halaman utama, yaitu halaman *login*, *katalog*, *rekomendasi*, dan *form* untuk menambahkan item ke *database*.

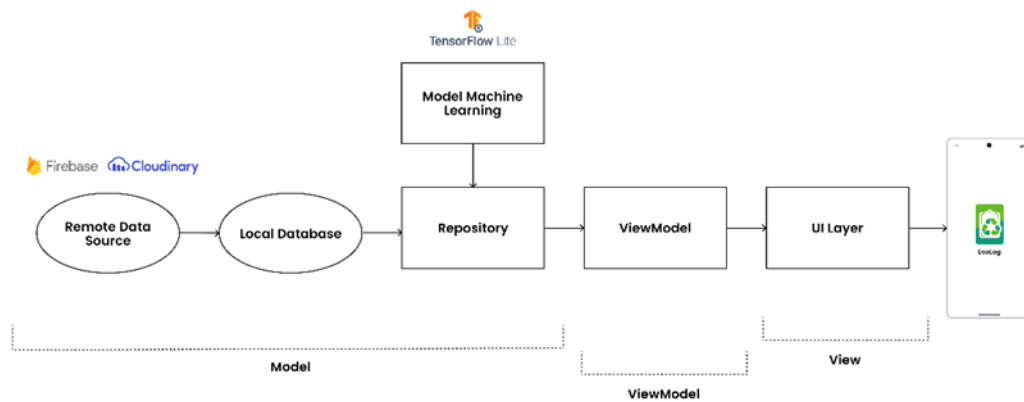
3. Pemrograman

Setelah desain aplikasi dan model *machine learning* selesai dibuat, langkah selanjutnya adalah mengimplementasikan desain tersebut menjadi sebuah program. Dimulai dari pemrograman model. Kemudian, dilanjutkan dengan pemrograman aplikasi dan integrasi model ke dalam aplikasi.

Model klasifikasi sampah dikembangkan menggunakan *transfer learning* dengan arsitektur *MobileNetV2*. Bagian atas model dimodifikasi dengan lapisan *Global Average Pooling*, dua lapisan *dense* (1024 dan 512 neuron dengan aktivasi *ReLU*), *dropout* (0.5 dan 0.3), serta *output layer* dengan 12 neuron dan aktivasi *softmax*, untuk mengklasifikasikan 2600 gambar ke dalam 12 kategori. Sebanyak 86 lapisan awal dibekukan guna mempertahankan bobot awal. Dataset diproses dengan *augmentasi* gambar menggunakan *ImageDataGenerator* (*rotasi*, *penggeseran*, *pembesaran*, *flipping*), dengan *input* berukuran 224x224 piksel, serta pembagian direktori pelatihan dan validasi. Model dikompilasi dengan *optimizer Adam* dengan *learning rate* 0,0001, fungsi *loss categorical crossentropy*, dan *class weight balancing* untuk mengatasi ketidakseimbangan kelas.

Pelatihan dilakukan menggunakan 50 *epoch*, dilanjutkan *fine-tuning* pada 20 lapisan akhir model dasar dengan *learning rate* lebih kecil (0,00001) selama 20 *epoch*. *Early stopping* dan model *checkpoint* digunakan untuk mencegah *overfitting* dan menyimpan model terbaik. Evaluasi dilakukan menggunakan *classification report* pada data validasi. Setelah pelatihan, model disimpan dalam format *.h5*, lalu dikonversi ke *.tflite* agar dapat digunakan secara lokal di aplikasi Android.





Gambar 7. Ilustrasi MVVM pada aplikasi

Pengembangan aplikasi *mobile*, atau *mobile development*, adalah proses menciptakan perangkat lunak yang secara khusus dirancang untuk beroperasi pada perangkat seluler seperti *smartphone* dan tablet [15]. Pengembangan aplikasi *mobile* dilakukan dengan menggunakan bahasa Kotlin dan arsitektur MVVM (Model, View, ViewModel) dengan menerapkan *repository* dan model untuk menangani *input* dan *output*, yang diilustrasikan pada gambar 7. Kunci dari penerapan arsitektur MVVM adalah menggunakan *Data Binding Library* untuk mewujudkan ikatan dua arah antara View dan ViewModel. Dalam struktur ini, ViewModel bertugas mengelola logika aplikasi dan status tampilan, Model berfungsi untuk menyimpan dan mengelola data bisnis, sedangkan View hanya bertanggung jawab untuk menampilkan data kepada pengguna [16]. Komponen seperti Firebase Authentication, Firestore, dan Cloudinary diintegrasikan untuk mengelola data pengguna, menyimpan informasi sampah, serta menangani unggahan gambar.

4. Pengujian Sistem

Proses pengujian untuk aplikasi dilakukan dalam dua tahap, yakni pengujian oleh pihak pengembang melalui metode *blackbox*, serta pengujian dari sisi pengguna yang dilakukan dengan evaluasi *usability* menggunakan kuesioner [17]. *Blackbox testing* adalah salah satu teknik dalam pengujian perangkat lunak yang digunakan untuk mengevaluasi fungsionalitas sebuah aplikasi. Metode ini berfokus pada pengujian terhadap *input* yang disediakan dalam aplikasi serta *output* yang diharapkan untuk setiap nilai *input* yang diberikan [18]. Selanjutnya, dilakukan pengukuran tingkat kepuasan dan kemudahan penggunaan aplikasi melalui *System Usability Scale* (SUS). SUS adalah instrumen evaluasi yang dikembangkan oleh John Brooke pada tahun 1996 untuk menilai kegunaan sistem melalui 10 item pertanyaan [19].

Selain pengujian aplikasi, dilakukan juga pengujian terhadap performa model *machine learning* menggunakan *classification report*. Hasil dari *classification report* ini yang merupakan tolok ukur kesiapan model untuk diimplementasikan.



5. Implementasi Sistem

Aplikasi ini ditujukan untuk individu atau rumah tangga yang menghasilkan sampah dalam jumlah kecil, seperti limbah rumah tangga atau sampah harian. Distribusi dilakukan melalui file APK yang dibagikan langsung kepada responden untuk diuji coba. Pada tahap ini juga, *model machine learning* diintegrasikan secara penuh ke dalam aplikasi untuk diuji dalam lingkungan nyata. Selanjutnya, untuk mengetahui *feedback* dari pengguna, kuesioner SUS dibagikan setelah mereka mencoba aplikasi.

6. Pemeliharaan Sistem

Pemeliharaan sistem merupakan aktivitas untuk merawat aplikasi melalui pembaruan dan peningkatan fitur secara berkala. Tujuannya adalah untuk memastikan aplikasi tetap bebas dari *bug*, serta melakukan penanganan apabila muncul masalah [20]. Jika ada laporan dari pengguna, pengembang akan menganalisis kesalahan yang terjadi, melakukan perbaikan, dan mendistribusikan kembali versi yang telah diperbaiki. Tahap ini dapat menjadi tolak ukur dalam keberlanjutan pengembangan aplikasi ini [21]

HASIL

Pada bab ini dijelaskan hasil dari implementasi dan pengujian sistem, baik dari sisi model klasifikasi gambar yang dibangun, maupun dari sisi aplikasi Android sebagai media pencatatan dan klasifikasi sampah. Pembahasan disusun dalam lima aspek utama sebagai berikut:

1. Hasil Evaluasi Model Klasifikasi (Classification Report)

Model klasifikasi dibangun menggunakan arsitektur *MobileNetV2* yang telah dilatih ulang (*transfer learning*). Dataset terdiri dari 12 kelas subkategori sampah. Training dilakukan dengan parameter *augmentasi data* (*rotasi, flipping, shifting*), optimisasi menggunakan Adam, serta *class weighting* untuk mengatasi ketidakseimbangan kelas. Evaluasi dilakukan menggunakan data validasi dan menghasilkan *classification report* seperti yang tertera pada tabel 1.

Tabel 1. *Classification Report* Model Klasifikasi Gambar

| Label Sampah | Precision | Recall | F1-Score | Support |
|--------------|-----------|--------|----------|---------|
| Botol | 0.91 | 0.98 | 0.94 | 50 |
| Kaca | 1.00 | 0.82 | 0.90 | 50 |
| Kaleng | 0.86 | 0.98 | 0.92 | 50 |



| Label Sampah | Precision | Recall | F1-Score | Support |
|---------------------------|-----------|--------|----------|---------|
| Kantong Plastik | 1.00 | 1.00 | 1.00 | 50 |
| Kardus | 0.96 | 0.96 | 0.96 | 50 |
| Kemasan Plastik | 1.00 | 0.92 | 0.96 | 50 |
| Kertas | 0.89 | 0.98 | 0.93 | 50 |
| Logam | 0.98 | 0.98 | 0.98 | 50 |
| Baterai dan Aki | 0.89 | 0.82 | 0.85 | 50 |
| Sampah Elektronik | 0.98 | 1.00 | 0.99 | 50 |
| Kayu | 0.94 | 0.96 | 0.95 | 50 |
| Sisa makanan dan dedaunan | 0.96 | 0.94 | 0.95 | 50 |
| Macro avg | 0.95 | 0.94 | 0.94 | |
| Weighted avg | 0.95 | 0.94 | 0.94 | |

Berdasarkan tabel 1, diperoleh nilai rata rata untuk *precision*, *recall*, dan *f1-score* adalah 0,95 ; 0,94 dan 0,94.

2. Evaluasi Implementasi Model dalam Aplikasi

Model hasil pelatihan kemudian dikonversi ke format TensorFlow Lite (.tflite) untuk digunakan dalam aplikasi Android. Pengujian lapangan dilakukan untuk mengukur performa model dalam konteks penggunaan nyata. Pengumpulan data dilakukan dengan melibatkan responden pengguna aplikasi untuk memindai berbagai jenis sampah secara langsung. Hasilnya adalah sebagai berikut:

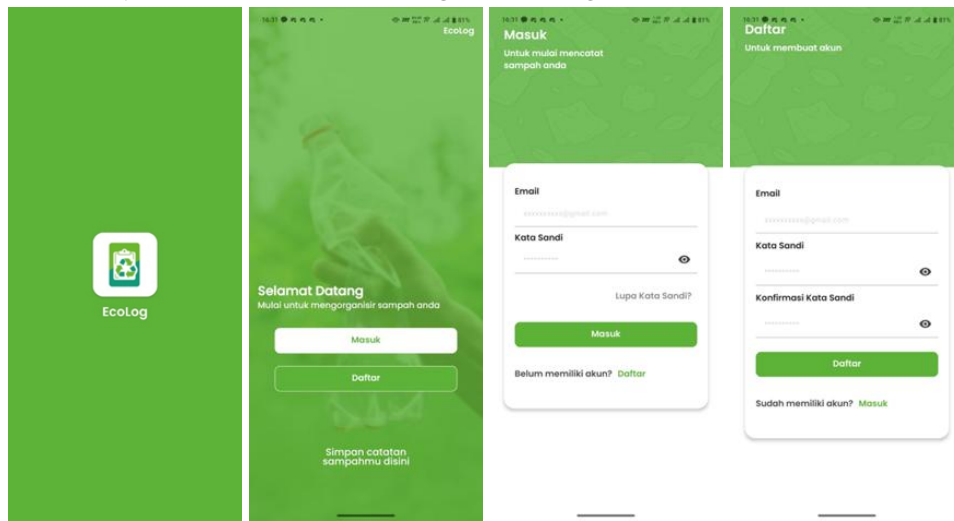
- Total sampah yang di-scan: 153
- Sampah berhasil diklasifikasi dengan benar: 128
- Sampah gagal atau salah klasifikasi: 25
- Akurasi klasifikasi di lingkungan nyata: Akurasi = $\frac{128}{153} \times 100\% \approx 83,66\%$

3. Tampilan Aplikasi

Berikut adalah tampilan dari aplikasi “EcoLog” yang dikembangkan dengan menggunakan bahasa pemrograman Kotlin dengan arsitektur MVVM.



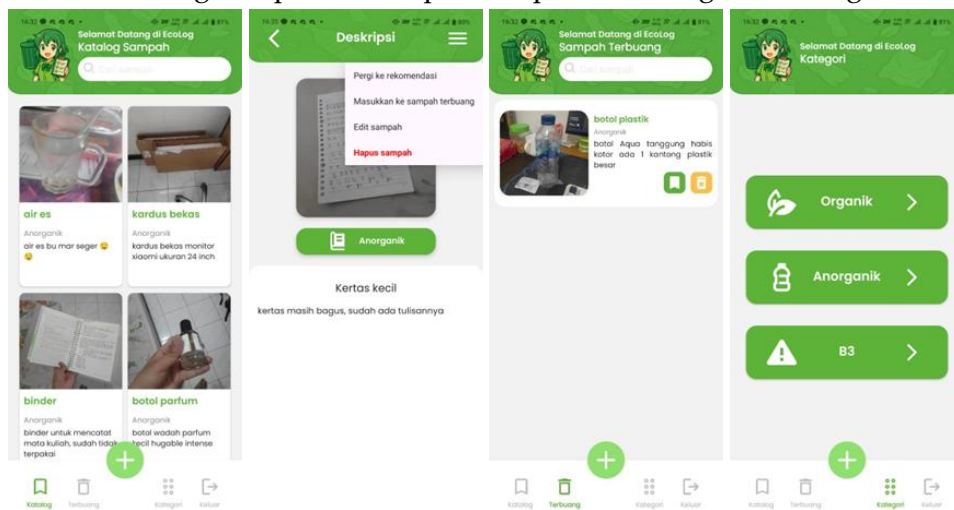
- Halaman *Splash Screen*, *Welcome*, *Login*, dan *Registrasi*



Gambar 8. UI splash screen, welcome, login, dan register

Pada gambar 8, dilampirkan tangkapan layar pada bagian *authentication* meliputi *splash screen*, *welcome*, *login*, dan *register*. Halaman *splash screen* merupakan halaman yang pertama kali akan dijalankan setelah pengguna masuk kedalam aplikasi. Pada halaman ini, terdapat logo dan nama aplikasi. Selanjutnya adalah halaman *welcome*, yang berisi tulisan selamat datang dan dua tombol untuk masuk atau daftar. Pada halaman *login*, pengguna dapat masuk ke aplikasi menggunakan email dan kata sandi. Jika pengguna belum memiliki akun maka pengguna dapat menuju ke halaman daftar. Pada halaman daftar, pengguna dapat membuat akun baru.

- Halaman *Katalog Sampah*, *Deskripsi*, *Sampah Terbuang*, dan *Kategori*

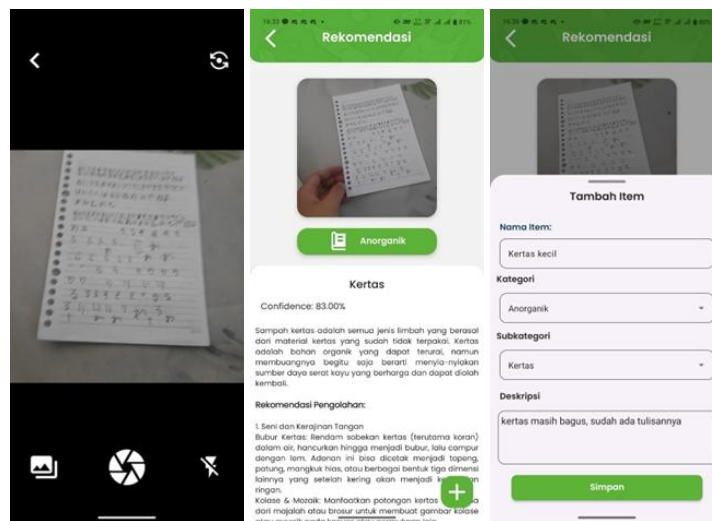


Gambar 9. UI katalog sampah, deskripsi, sampah terbuang, dan kategori



Tampilan halaman utama aplikasi tercantum pada gambar 9. Dimulai dari halaman katalog sampah yang merupakan halaman utama setelah pengguna berhasil masuk. Halaman ini menampilkan daftar sampah yang telah discan dan disimpan oleh pengguna. Saat salah satu item sampah diklik, maka akan pergi ke halaman deskripsi. Halaman deskripsi berisi detail dari sampah yang telah disimpan. Disini pengguna juga bisa masuk ke rekomendasi pengolahan, memasukkan sampah ke daftar terbuang, mengedit sampah yang telah disimpan, dan menghapus sampah dari data. Halaman sampah terbuang berfungsi untuk memisahkan sampah aktif dari sampah yang telah dibuang oleh pengguna tanpa menghapusnya dari *database*. Fungsi ini berguna ketika pengguna ingin melakukan pengolahan terhadap sampah mereka, namun tidak ingin menghapusnya dari catatan. Berikutnya halaman kategori, berisi setiap kategori sampah yang dapat discan dan diklasifikasi menggunakan aplikasi ini. Setiap kategori ketika diklik akan muncul rekomendasi pengolahannya. Jadi pengguna dapat membaca rekomendasi pengolahan sampah sesuai dengan kategorinya tanpa perlu scan sampah terlebih dahulu.

- Halaman Kamera, Rekomendasi, dan Form Tambah Data



Gambar 10. UI kamera, rekomendasi, dan *form* tambah data

Pada gambar 10, terdapat tampilan dari bagian aplikasi yang mengurus klasifikasi sampah dan menyimpan sampah ke *database*. Halaman kamera digunakan untuk memotret gambar sampah atau memilih gambar dari galeri untuk sampah. Setelah gambar berhasil diambil, pengguna akan diarahkan ke halaman rekomendasi pengolahan. Pada halaman ini, sampah diklasifikasikan sesuai dengan kategorinya, untuk kemudian diberikan rekomendasi pengolahan. Selanjutnya adalah *form* tambah data yang akan muncul ketika pengguna menekan tombol tambah yang ada di halaman rekomendasi. Pada bagian

pengguna dapat menambahkan sampah sesuai dengan hasil gambar yang telah diambil. Setelah mengisi semua data, sampah akan disimpan pada *database* dan ditampilkan di halaman katalog sampah.

4. Pengujian Fungsional (Blackbox Testing)

Pengujian fungsional dilakukan menggunakan metode *blackbox testing*, yaitu pendekatan pengujian berdasarkan hasil keluaran (*output*) terhadap masukan (*input*) tanpa memeriksa kode sumber aplikasi secara langsung. Tujuan dari pengujian ini adalah untuk memastikan bahwa setiap fitur utama dalam aplikasi bekerja sesuai dengan spesifikasi fungsional yang telah dirancang. Ringkasan pengujian *blackbox testing* yang dilakukan dilampirkan dalam tabel 2.

Tabel 2. Hasil pengujian *blackbox*

| No | Nama Fitur | Input Uji | Hasil yang Diharapkan | Status |
|----|--|----------------------------------|--|----------|
| 1 | Login | Email dan Password Valid | Pengguna Berhasil Login | Berhasil |
| 2 | Daftar | Form Valid | Akun Berhasil Dibuat | Berhasil |
| 3 | Lupa Sandi | Email Valid dan Sandi Baru | Sandi berhasil diubah sesuai email yang dimasukkan | Berhasil |
| 4 | Klasifikasi Gambar | Gambar Sampah | Label sesuai hasil model | Berhasil |
| 5 | Tambah Data Sampah | Form Lengkap dan Gambar | Data tersimpan di database | Berhasil |
| 6 | Edit, Hapus, dan Pindahkan Data Ke Daftar Terbuang | Klik Edit, Hapus, atau Pindahkan | Data diperbarui atau dihapus | Berhasil |
| 7 | Navigasi Antar Halaman | Klik Setiap Menu Navigasi | Halaman tampil sesuai navigasi | Berhasil |

Berdasarkan tabel 2, dapat disimpulkan bahwa seluruh fungsi utama yang diujikan pada pengujian *blackbox* berjalan sesuai perancangan.



5. Evaluasi Penggunaan Aplikasi Menggunakan SUS (System Usability Scale)

Untuk mengevaluasi usability aplikasi, dilakukan menggunakan metode *System Usability Scale* (SUS) terhadap 30 responden, setelah mereka mencoba langsung menggunakan aplikasi untuk login, scan gambar sampah, menyimpan data, dan melihat katalog. Berikut daftar pertanyaan yang ditanyakan kepada responden dalam mengimplementasikan SUS pada tabel 3.

Tabel 3. Daftar pertanyaan SUS

| No | Pertanyaan |
|----|--|
| 1 | Saya merasa akan sering menggunakan aplikasi ini. |
| 2 | Saya merasa aplikasi ini terlalu kompleks padahal dapat dibuat secara lebih sederhana. |
| 3 | Saya merasa aplikasi ini mudah digunakan. |
| 4 | Saya membutuhkan bantuan teknis untuk dapat menggunakan aplikasi ini |
| 5 | Fitur-fitur yang ada di aplikasi ini sudah terintegrasi dengan baik sesuai dengan fungsinya. |
| 6 | Saya menemukan banyak hal yang menurut saya tidak konsisten pada aplikasi ini. |
| 7 | Saya rasa mayoritas pengguna akan dapat belajar menggunakan sistem ini dengan cepat. |
| 8 | Aplikasi ini terasa membingungkan saat digunakan. |
| 9 | Saya merasa sangat percaya diri untuk bisa mengoperasikan aplikasi ini. |
| 10 | Saya harus banyak belajar sebelum bisa menggunakan aplikasi ini. |

Sesuai tabel 3, dari 10 pertanyaan yang ditanyakan, kami memberikan 5 opsi respon yang menyatakan kesetujuan atau ketidaksetujuan dalam skala likert 5 poin (1 = sangat tidak setuju; 5 sangat setuju). Perhitungan skor SUS dilakukan dengan menjumlahkan skor setiap pertanyaan. Setiap pertanyaan memiliki rentang skor antara 0 sampai 4. Untuk pertanyaan ganjil, skornya dihitung dengan cara mengurangi 1 dari nilai skala yang dipilih oleh responden. Sedangkan untuk pertanyaan genap, skornya dihitung dengan mengurangi nilai skala dari angka 5. Setelah semua skor dijumlahkan, hasil total tersebut dikalikan dengan 2,5 untuk



mendapat skor akhir SUS [19]. Tabel hasil SUS yang telah kami jalankan, kami rangkum dalam tabel 4 berikut.

Tabel 4. Hasil SUS

| Responden | Pertanyaan | | | | | | | | | | Jumlah | Hasil (kali 2,5) |
|-----------|------------|---|---|---|---|---|---|---|---|----|--------|---------------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | |
| 1 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 40 | 100 |
| 2 | 3 | 4 | 4 | 4 | 4 | 3 | 4 | 4 | 4 | 4 | 38 | 95 |
| 3 | 2 | 4 | 4 | 4 | 2 | 3 | 4 | 4 | 4 | 4 | 35 | 87,5 |
| 4 | 0 | 4 | 4 | 4 | 3 | 4 | 4 | 4 | 4 | 4 | 35 | 87,5 |
| 5 | 3 | 4 | 4 | 2 | 4 | 3 | 4 | 3 | 4 | 4 | 35 | 87,5 |
| 6 | 0 | 1 | 4 | 4 | 2 | 3 | 3 | 4 | 3 | 4 | 28 | 70 |
| 7 | 4 | 4 | 4 | 4 | 4 | 3 | 4 | 4 | 4 | 4 | 39 | 97,5 |
| 8 | 2 | 3 | 3 | 4 | 3 | 3 | 3 | 3 | 2 | 3 | 29 | 72,5 |
| 9 | 3 | 2 | 4 | 3 | 3 | 2 | 4 | 2 | 4 | 3 | 30 | 75 |
| 10 | 3 | 4 | 4 | 3 | 3 | 4 | 4 | 3 | 4 | 4 | 36 | 90 |
| 11 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 4 | 4 | 3 | 36 | 90 |
| 12 | 3 | 4 | 4 | 4 | 4 | 4 | 3 | 4 | 3 | 4 | 37 | 92,5 |
| 13 | 3 | 2 | 3 | 1 | 4 | 3 | 4 | 4 | 4 | 4 | 32 | 80 |
| 14 | 1 | 2 | 3 | 4 | 4 | 3 | 4 | 3 | 3 | 3 | 30 | 75 |
| 15 | 4 | 4 | 4 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 38 | 95 |
| 16 | 4 | 3 | 4 | 3 | 3 | 2 | 4 | 4 | 3 | 4 | 34 | 85 |
| 17 | 2 | 4 | 4 | 3 | 3 | 2 | 4 | 4 | 4 | 4 | 34 | 85 |
| 18 | 2 | 4 | 4 | 4 | 3 | 3 | 3 | 4 | 4 | 4 | 35 | 87,5 |
| 19 | 2 | 1 | 3 | 2 | 3 | 2 | 2 | 1 | 3 | 2 | 21 | 52,5 |
| 20 | 4 | 2 | 4 | 2 | 4 | 3 | 4 | 3 | 4 | 3 | 33 | 82,5 |



| Responden | Pertanyaan | | | | | | | | | | Jumlah | Hasil (kali 2,5) |
|-----------|------------|---|---|---|---|---|---|---|---|----|--------|---------------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | |
| 21 | 4 | 0 | 1 | 1 | 3 | 4 | 1 | 1 | 1 | 3 | 19 | 47,5 |
| 22 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 39 | 97,5 |
| 23 | 4 | 2 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 38 | 95 |
| 24 | 4 | 4 | 4 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 39 | 97,5 |
| 25 | 3 | 1 | 3 | 1 | 3 | 1 | 3 | 1 | 3 | 1 | 20 | 50 |
| 26 | 2 | 3 | 3 | 4 | 2 | 3 | 4 | 3 | 3 | 2 | 29 | 72,5 |
| 27 | 4 | 3 | 4 | 2 | 3 | 2 | 4 | 4 | 4 | 2 | 32 | 80 |
| 28 | 4 | 4 | 3 | 4 | 4 | 3 | 4 | 4 | 4 | 3 | 37 | 92,5 |
| 29 | 4 | 2 | 4 | 2 | 4 | 3 | 4 | 4 | 4 | 1 | 32 | 80 |
| 30 | 4 | 0 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 36 | 90 |
| Rata-rata | | | | | | | | | | | 33,2 | 83 |

Berdasarkan tabel 4 diatas, diperoleh hasil akhir berupa rata-rata dari skor SUS adalah 83.

PEMBAHASAN

Bagian ini membahas secara mendalam mengenai hasil dari penelitian, mulai dari evaluasi performa model sampai dengan pengujian aplikasi.

1. Evaluasi Implementasi Model dalam Aplikasi *Mobile*

Model yang telah dilatih dan divalidasi dengan data validasi, kemudian dikonversi ke dalam format TensorFlow Lite (.tflite). Model inilah yang digunakan didalam aplikasi *mobile* untuk mengklasifikasikan sampah pada lingkungan nyata. Pada tabel 5, disajikan ringkasan aspek evaluasi dan nilai dari model terhadap pengetesan di lingkungan nyata.



Tabel 5. Evaluasi implementasi model pada lingkungan nyata

| Aspek Evaluasi | Nilai |
|----------------------------------|---|
| Akurasi data validasi | $\pm 94-95\%$ |
| Akurasi implementasi di aplikasi | $\pm 84\%$ |
| Perbedaan Akurasi | $\pm 10-11\%$ |
| Interpretasi | Model bekerja baik di lingkungan terkontrol, dan cukup stabil di lingkungan nyata, namun masih perlu peningkatan robustness terhadap variasi <i>input</i> . |

Pada tabel 5 menunjukkan bahwa meskipun model mencapai tingkat akurasi tertentu saat diuji di lingkungan pengembangan, hasil implementasinya dalam aplikasi Android dapat mengalami sedikit penurunan atau pergeseran prediksi. Hal ini dapat disebabkan oleh perbedaan lingkungan, *preprocessing input*, maupun kendala perangkat keras.

Dengan membandingkan hasil ini dengan literatur sejenis, penelitian ini memperlihatkan bahwa implementasi nyata model AI pada perangkat *mobile* tetap mampu mempertahankan performa yang layak untuk kebutuhan klasifikasi sampah berbasis kategori. Hasil ini mengindikasikan bahwa meskipun terdapat penurunan minor, aplikasi tetap layak digunakan oleh pengguna umum.

2. Evaluasi Desain Antarmuka Aplikasi

Evaluasi desain antarmuka difokuskan pada keterpakaian, kemudahan navigasi, dan konsistensi elemen visual dalam aplikasi. Antarmuka pengguna merupakan aspek krusial dalam menentukan sejauh mana aplikasi dapat diterima dan digunakan oleh masyarakat secara luas, terutama dalam konteks lingkungan rumah tangga yang heterogen.

Dari hasil observasi dan umpan balik pengguna, sebagian besar responden menyatakan bahwa aplikasi mudah digunakan, dengan alur navigasi yang jelas dan ikon yang representatif. Namun, terdapat beberapa catatan terkait konten dari aplikasi. Misalnya, beberapa pengguna dengan rentang usia remaja menuju dewasa (15-21 tahun) mengeluhkan terkait tampilan aplikasi yang minim konten dan terlalu monoton, yang didominasi oleh tulisan sehingga mereka cepat merasa bosan. Namun, pada pengguna dengan rentang usia lebih tinggi, merasa bahwa aplikasi ini sudah



cukup sesuai. Karena fitur utamanya yaitu untuk pencatatan dan klasifikasi sampah sudah ada dan cukup mudah untuk digunakan.

Dari umpan balik yang didapatkan, bisa disimpulkan bahwa aplikasi ini mudah digunakan oleh berbagai kalangan, termasuk remaja sampai dengan ibu rumah tangga. Namun, konten didalamnya masih terlalu monoton sehingga perlu menambahkan beberapa hal seperti gambar/video terkait rekomendasi pengolahan, atau menerapkan sistem poin dan gamifikasi untuk setiap sampah yang telah diklasifikasi. Hal ini dapat dilakukan untuk menarik lebih banyak pengguna untuk mencoba aplikasi ini.

3. Analisis Hasil Pengujian Fungsional

Pengujian fungsional dilakukan dengan pendekatan *blackbox testing* untuk memastikan bahwa setiap fitur dalam aplikasi bekerja sesuai dengan skenario penggunaan yang telah dirancang. Kami menjalankan pengujian *blackbox testing* terhadap 7 fitur utama dari aplikasi. Sesuai dengan alur penggunaan aplikasi, kami menguji fitur dimulai dari fitur *login*, daftar, dan lupa sandi yang ketiganya sama-sama berhubungan dengan *authentication* menggunakan firebase auth. Kemudian, kami melanjutkan ke pengujian navigasi antar halaman. Selanjutnya, kami menjalankan pengujian untuk scan gambar dari kamera dan galeri untuk dilakukan pengklasifikasian sampah. Terakhir, kami melakukan pengujian untuk menambah, mengedit, menghapus, dan memindahkan data sampah.

Hasil pengujian menunjukkan bahwa seluruh fungsionalitas utama berjalan sesuai ekspektasi. Tidak ditemukan *error* atau *i* yang dapat mengganggu fungsi utama dari aplikasi. Hal ini menunjukkan bahwa aplikasi telah siap dibagikan kepada pengguna di lingkungan nyata.

Keterbatasan dalam pengujian ini adalah bahwa pengujian dilakukan dalam skala terbatas, sehingga tidak sepenuhnya merepresentasikan keragaman perangkat dan kondisi lingkungan pengguna di lapangan. Namun demikian, aplikasi ini menunjukkan potensi tinggi untuk digunakan secara luas dengan pengujian lanjutan berskala lebih besar.

4. Analisis Hasil Penggunaan Aplikasi (SUS)

Analisis ini dilakukan untuk mengevaluasi sejauh mana aplikasi diterima oleh pengguna dari segi *usability*, dengan menggunakan metode *System Usability Scale* (SUS).





Gambar 11. Rentang skor SUS

Pada gambar 11 dijelaskan proses penentuan hasil akhir SUS berdasarkan tiga aspek utama yaitu *acceptability*, *grade scale*, dan *adjective rating*. *Acceptability* digunakan untuk menilai sejauh mana aplikasi dapat diterima, *Grade scale* merepresentasikan tingkat kualitas aplikasi, *adjective rating* digunakan untuk memberikan penilaian subjektif terhadap seberapa bermanfaat aplikasi [22]. Rentang skor SUS dijelaskan secara rinci dalam gambar 11, skor di bawah 40 dikategorikan sebagai *not acceptable* yang menunjukkan bahwa aplikasi belum layak dipublikasikan atau digunakan. Skor antara 40 hingga 60 masuk dalam kategori *marginal* yang berarti beberapa fitur sudah dapat digunakan tetapi aplikasi masih memerlukan banyak penyempurnaan. Adapun skor di atas 60 hingga 100 menunjukkan bahwa aplikasi telah memenuhi standar kelayakan dan dinyatakan lolos uji kelayakan SUS [23].

Berdasarkan data yang diperoleh, skor rata-rata yang didapatkan dari hasil SUS yang diujikan pada aplikasi ini adalah 83. Dilihat dari rentang penilaian hasil SUS, aplikasi ini masuk ke dalam kategori *acceptable*, dengan *adjective excellent hingga best imaginable* dan memiliki *grade A*. Artinya, aplikasi ini memiliki tingkat *usability* yang tinggi, dengan persepsi pengguna terhadap kemudahan penggunaan, kecepatan belajar, dan kenyamanan antarmuka berada pada tingkat memuaskan. Meskipun begitu, perlu diingat bahwa responden yang mengisi pertanyaan ini bervariasi dan kami tidak dapat mengatur apakah responden tersebut benar-benar telah mencoba aplikasinya atau belum. Hal ini dapat mempengaruhi hasil akhir dari skor atau nilai SUS.

KESIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa pengembangan aplikasi pencatatan dan klasifikasi sampah berbasis kecerdasan buatan berhasil dilakukan dengan pendekatan rekayasa perangkat lunak *waterfall*. Model *MobileNetV2* yang digunakan mampu dikonversi ke format TensorFlow Lite dan diintegrasikan ke dalam aplikasi Android dengan hasil evaluasi yang baik, mencatat nilai *presisi* 95%, *recall* 94%, dan *F1-score* 94%. Dalam pengujian nyata, model menunjukkan tingkat keberhasilan klasifikasi



sebesar 84% terhadap 153 sampel gambar dari kamera secara langsung. Seluruh fitur utama aplikasi termasuk login, klasifikasi, pengelompokan kategori, dan penyimpanan data berfungsi sesuai perancangan, sebagaimana dibuktikan oleh pengujian *blackbox* pada berbagai perangkat Android. Evaluasi *usability* menggunakan *System Usability Scale* (SUS) terhadap 30 responden menghasilkan skor rata-rata 83 yang tergolong kategori "Excellent", menandakan bahwa aplikasi ini mudah digunakan dan diterima baik oleh pengguna. Ke depannya, aplikasi ini masih dapat dikembangkan lebih lanjut, seperti peningkatan akurasi dan cakupan klasifikasi model, penambahan fitur sosial, *gamifikasi*, integrasi lokasi, kerja sama dengan instansi terkait, serta penyediaan *onboarding tutorial* untuk membantu pengguna baru.

UCAPAN TERIMA KASIH

Ucapan Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan tugas akhir yang berjudul "Rancang Bangun Aplikasi Katalog dan Klasifikasi Sampah Rumah Tangga Berbasis Android Menggunakan Kotlin dan *MobileNet*" ini dengan baik.

Penulis menyadari bahwa tersusunnya tugas akhir ini tidak lepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua tercinta, yang selalu memberikan doa, dukungan moral, dan semangat tanpa henti dalam setiap langkah yang penulis tempuh.
2. Bapak Endang Wahyu Pamungkas, S.Kom., M.Kom., Ph.D. selaku dosen pembimbing yang telah memberikan arahan, masukan, serta motivasi dalam penyusunan dan penyempurnaan tugas akhir ini.
3. Ibu Dr. Endah Sudarmilah, S.T., M.Eng. dan Bapak Dedi Gunawan, S.T., M.Sc., Ph.D. selaku dekan Fakultas Komunikasi dan Informatika dan kaprodi Teknik Informatik Universitas Muhammadiyah Surakarta yang telah memberikan kesempatan dan fasilitas selama proses perkuliahan hingga penyusunan tugas akhir.
4. Seluruh dosen pengajar dan staf akademik di lingkungan Fakultas Komunikasi dan Informatika Universitas Muhammadiyah Surakarta, yang telah membekali penulis dengan ilmu pengetahuan dan pengalaman berharga.
5. Teman-teman seperjuangan yang telah menjadi tempat berbagi semangat dan saling membantu selama proses pengerjaan tugas akhir.



6. Semua pihak yang tidak dapat penulis sebutkan satu per satu, namun telah memberikan bantuan baik secara langsung maupun tidak langsung dalam penyusunan tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna. Namun, penulis berharap karya ini dapat memberikan manfaat dan menjadi sumbangsih kecil bagi pengembangan ilmu pengetahuan.

DAFTAR PUSTAKA

- [1] United Nations Department of Economic and Social Affairs, "World Population Prospects (2024)," United Nations. Accessed: Mar. 18, 2025. [Online]. Available: <https://ourworldindata.org/grapher/population-unwpp?time=latest&country=CHN~IND~USA~IDN>
- [2] MENLHK, "Capaian Kinerja Pengelolaan Sampah." Accessed: Mar. 12, 2025. [Online]. Available: <https://sipsn.menlhk.go.id/sipsn/>
- [3] P. T. Violetta, "Pemerintah targetkan pengelolaan sampah 100 persen pada 2029." Accessed: Mar. 13, 2025. [Online]. Available: <https://www.antaraneews.com/berita/4700097/pemerintah-targetkan-pengelolaan-sampah-100-persen-pada-2029>
- [4] E. Damanhuri and T. Padmi, "Pengelolaan sampah," Diktat kuliah TL, vol. 3104, pp. 5–10, 2010.
- [5] A. Ratnasari, I. S. Asharhani, M. G. Sari, S. R. Hale, and H. Pratiwi, "Edukasi pemilahan sampah sebagai upaya preventif mengatasi masalah sampah di lingkungan sekolah," Prosiding Konferensi Nasional Pengabdian Kepada Masyarakat Dan Corporate Social Responsibility (PKM-CSR), vol. 2, pp. 652–659, 2019.
- [6] Y. Bassil, "A Simulation Model for the Waterfall Software Development Life Cycle," 2012. [Online]. Available: <https://arxiv.org/abs/1205.6904>
- [7] W. Gunawan and B. Seno, "Implementasi Algoritma Fuzzy C-Means Clustering Sistem Crowdfunding pada Sektor Industri Kreatif Berbasis Web," Jurnal Edukasi dan Penelitian Informatika (JEPIN), vol. 6, pp. 193–201, Aug. 2020, doi: 10.26418/jp.v6i2.38018.
- [8] S. Laato, M. Mäntymäki, M. Minkkinen, T. Birkstedt, A. K. M. Islam, and D. Dennehy, Integrating *Machine learning* With Software Development Lifecycles: Insights From Experts. 2022.



- [9] D. Ardianto, Y. M. Juliansyah, D. Ramayanti, and others, "Design And Development Of A Web-Based Posyandu Application Using The Waterfall Method," *Instal: Jurnal Komputer*, vol. 15, no. 02, pp. 309–319, 2023.
- [10] D. Andrian, "Penerapan Metode Waterfall dalam Perancangan Sistem Informasi Pengawasan Proyek Berbasis Web," 2021.
- [11] A. Gemino and D. Parker, "Use Case Diagrams in Support of Use Case Modeling: Deriving Understanding from the Picture," *Journal of Database Management (JDM)*, vol. 20, no. 1, pp. 1–24, 2009, doi: 10.4018/jdm.2009010101.
- [12] M. Dumas and A. H. M. Ter Hofstede, "UML activity diagrams as a workflow specification language," in *International conference on the unified modeling language*, Springer, 2001, pp. 76–90.
- [13] P. Chen, "Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned," Mar. 2002, doi: 10.1007/978-3-642-59412-0_17.
- [14] N. Ngo, "Android software development: Case: 'Logo Quiz World' Mobile Application," 2019.
- [15] S. P. Simbolon and R. Maulany, "Perancangan Aplikasi Pendeteksi Jenis-jenis Sampah Berbasis Android: Development of an Android-Based Waste Type Detection," *MALCOM: Indonesian Journal of Machine learning and Computer Science*, vol. 4, no. 3, pp. 926–935, May 2024, doi: 10.57152/malcom.v4i3.1337.
- [16] W. Sun, H. Chen, and W. Yu, "The Exploration and Practice of MVVM Pattern on Android Platform," in *Proceedings of the 2016 4th International Conference on Machinery, Materials and Information Technology Applications*, Atlantis Press, Jan. 2017. doi: 10.2991/icmmita-16.2016.205.
- [17] F. Y. Al Irsyadi, R. Annas, and Y. I. Kurniawan, "Game Edukasi Pembelajaran Bahasa Inggris untuk Pengenalan Benda-Benda di Rumah bagi Siswa Kelas 4 Sekolah Dasar," *Jurnal Teknologi dan Informasi*, vol. 9, no. 2, pp. 78–92, Aug. 2019, doi: 10.34010/jati.v9i2.1844.
- [18] A. Verma, A. Khatana, and S. Chaudhary, "A Comparative Study of Black Box Testing and White Box Testing," *International Journal of Computer Sciences and Engineering*, vol. 5, pp. 301–304, Dec. 2017, doi: 10.26438/ijcse/v5i12.301304.
- [19] J. Brooke, "SUS -- a quick and dirty usability scale," 1996, pp. 189–194.



- [20] Maryam and H. W. Aryono, "Sistem Pakar Pengklasifikasi Stadium Kanker Serviks Berbasis *Mobile* Menggunakan Metode Decision Tree," *Jurnal Kajian Ilmiah*, vol. 22, no. 3, pp. 267–278, May 2024, doi: 10.31599/an383m44.
- [21] L. P. Arista and Y. S. Nugroho, "Sistem Informasi Pencatatan Transaksi Penjualan dan Pembelian Produk Berbasis Website di Toko Sembako Putrasena Sukoharjo," *Jurnal Informatika Polinema*, vol. 9, no. 4, pp. 397–404, Aug. 2023, doi: 10.33795/jip.v9i4.1347.
- [22] A. Fatmawati, "Evaluasi Usability pada Learning Management System OpenLearning Menggunakan System Usability Scale," *INOVTEK Polbeng - Seri Informatika*, vol. 6, p. 120, Jun. 2021, doi: 10.35314/isi.v6i1.1881.
- [23] A. Bangor, P. Kortum, and J. Miller, "Determining what individual SUS scores mean: adding an adjective rating scale," *J. Usability Studies*, vol. 4, no. 3, pp. 114–123, May 2009.

