

BAB I

PENDAHULUAN

A. Latar Belakang

Sampah merupakan salah satu permasalahan lingkungan global yang terus meningkat dari tahun ke tahun. Menurut data Bank Dunia (*The World Bank*), jumlah sampah di Indonesia pada tahun 2025 diperkirakan mencapai 67 juta ton atau setara dengan 175 ribu ton per hari. Jumlah tersebut tentunya akan semakin membebani petugas kebersihan dalam mengolah dan memilah jenis sampah secara manual.

Pengelolaan sampah merupakan salah satu tantangan utama yang dihadapi oleh masyarakat modern. Dalam era yang semakin berurbanisasi dan konsumerisme yang meningkat, jumlah sampah yang dihasilkan terus bertambah dengan cepat. Penanganan yang tidak efektif terhadap sampah dapat mengakibatkan berbagai masalah lingkungan, kesehatan, dan keberlanjutan.

Salah satu aspek penting dalam pengelolaan sampah adalah pengelompokan dan pemilahan sampah berdasarkan jenisnya. Pemilahan yang benar memungkinkan sampah untuk diolah kembali, didaur ulang, atau dibuang dengan cara yang sesuai, yang dapat mengurangi dampak negatif terhadap lingkungan.

Namun, pemilahan sampah yang dilakukan oleh masyarakat sering kali tidak akurat atau tidak konsisten. Hal ini bisa disebabkan oleh kurangnya pengetahuan tentang jenis sampah, kebingungan dalam mengidentifikasi jenis yang tepat, atau kesulitan dalam menghafal panduan pemilahan yang rumit. Akibatnya, sampah

yang seharusnya dapat didaur ulang atau diolah kembali seringkali berakhir di tempat pembuangan akhir, menyebabkan penumpukan sampah dan pemborosan sumber daya.

Oleh karena itu, dibutuhkan suatu sistem otomatis yang dapat mendeteksi jenis sampah secara cepat dan akurat tanpa harus melibatkan banyak tenaga manusia. Salah satu solusi teknologi yang dapat digunakan untuk menjawab tantangan ini adalah *machine learning*.

Menurut penelitian Alden et al. (2023) tentang penggunaan *machine learning* pada deteksi jenis sampah, hasil percobaan menunjukkan bahwa teknologi ini mampu memberikan tingkat akurasi sebesar 92% dalam pemilahan sampah organik dan anorganik serta kertas/karton dan plastik[1].

Selain itu, implementasi algoritma klasifikasi juga telah berhasil diterapkan pada berbagai bidang termasuk dalam industri rekayasa bahan bakar nuklir seperti penelitian Joan Yuliana Hutapea (2021) tentang pendugaan harga saham menggunakan metode *decision tree classification algorithm*[2].

Melihat potensi dari penggunaan *machine learning* dan algoritma klasifikasi pada deteksi jenis sampah, penulis merasa perlu melakukan penelitian tentang implementasi teknologi tersebut guna mendapatkan hasil yang lebih optimal dalam pemilahan sampah.

Dengan adanya implementasi *machine learning* menggunakan algoritma klasifikasi untuk mendeteksi jenis sampah, diharapkan pengelolaan sampah dapat ditingkatkan secara keseluruhan. Dengan adanya sistem yang dapat mengidentifikasi jenis sampah secara otomatis, pengguna dapat dengan mudah

memilah sampah dengan benar, yang pada gilirannya dapat mengurangi jumlah sampah yang berakhir di tempat pembuangan akhir dan meningkatkan upaya daur ulang serta pengelolaan sampah secara berkelanjutan.

B. Rumusan Masalah

Berdasarkan uraian dari latar belakang diatas, peneliti dapat mengambil rumusan masalah yang akan dilakukan yaitu

1. Bagaimana mengidentifikasi jenis sampah secara akurat menggunakan teknik *machine learning* dan algoritma klasifikasi *random forest*?
2. Bagaimana mengumpulkan dan mempersiapkan dataset yang mencakup berbagai jenis sampah untuk melatih model *machine learning*?
3. Bagaimana melakukan pelatihan model *machine learning* menggunakan algoritma *random forest* untuk mendeteksi jenis sampah?
4. Bagaimana mengukur dan mengevaluasi kinerja model *machine learning* dalam mendeteksi jenis sampah?

C. Tujuan Penelitian

1. Untuk mengidentifikasi jenis sampah secara akurat menggunakan teknik *machine learning* dan algoritma klasifikasi *random forest*.
2. Untuk mengumpulkan dan mempersiapkan dataset yang mencakup berbagai jenis sampah untuk melatih model *machine learning*.
3. Untuk melakukan pelatihan model *machine learning* menggunakan algoritma *random forest* untuk mendeteksi jenis sampah.
4. Untuk mengukur dan mengevaluasi kinerja model *machine learning* dalam mendeteksi jenis sampah.

D. Batasan Masalah

Batasan masalah dalam hal ini adalah deteksi jenis sampah yang terbatas pada organik (seperti sisa makanan, daun, atau serpihan kayu) dan anorganik (seperti plastik, logam, kertas, atau kaca). Jenis sampah lainnya, seperti bahan berbahaya, limbah medis, atau elektronik, tidak termasuk dalam ruang lingkup deteksi ini. Fokus deteksi jenis sampah akan diberikan pada kedua kategori tersebut, dengan tujuan mengklasifikasikan sampah berdasarkan atribut-atribut yang relevan.

E. Manfaat Penelitian

Manfaat dari aplikasi pendeteksi jenis sampah dengan metode *machine learning* yaitu,

a. Manfaat bagi Penulis

Penulis akan terlibat dalam pemahaman yang mendalam tentang konsep dan aplikasi *machine learning*, khususnya algoritma klasifikasi *Random Forest*.

b. Manfaat bagi Pengguna

Dengan menggunakan teknik *machine learning* dan algoritma klasifikasi, pengelolaan sampah dapat dilakukan secara lebih efisien. Proses identifikasi dan pemilahan sampah dapat diotomatisasi, mengurangi waktu dan tenaga yang diperlukan dalam proses tersebut.

c. Manfaat bagi Universitas

Penelitian ini melibatkan aspek-aspek dari bidang *machine learning* dan pengelolaan sampah, sehingga dapat mendorong kolaborasi antara fakultas atau departemen yang berbeda di dalam universitas. Hal ini dapat

merangsang penelitian lintas disiplin dan memperluas cakupan pengetahuan dan pemahaman di antara fakultas atau departemen yang terlibat.

BAB II

TINJAUAN PUSTAKA

A. Penelitian Terdahulu

Pengelolaan sampah yang efisien dan berkelanjutan menjadi tantangan penting dalam menjaga kelestarian lingkungan. Dalam beberapa tahun terakhir, penggunaan teknik machine learning dan algoritma klasifikasi telah menjadi fokus penelitian untuk mengidentifikasi jenis sampah dengan akurasi tinggi. Dalam tinjauan pustaka ini, akan dibahas penelitian terkait yang telah dilakukan menggunakan *machine learning* dan algoritma klasifikasi *Random Forest* untuk tujuan identifikasi jenis sampah.

Penelitian sebelumnya dilakukan oleh Agus Setiawan dengan judul “*Algoritma Decision Tree C4.5 Untuk Klasifikasi Tingkat Pengetahuan Masyarakat Terhadap Pemanfaatan Bank Sampah Di Kabupaten Banjarmasin Selatan*”[3]. Penelitian ini bertujuan untuk meningkatkan pemanfaatan Bank Sampah dengan menerapkan metode data mining menggunakan *Algoritma Decision Tree C4.5* berdasarkan faktor pengetahuan, pendidikan, dan usia masyarakat. Hasil pengujian menunjukkan bahwa model yang dikembangkan menggunakan metode data mining dan *Algoritma Decision Tree C4.5* mencapai akurasi sebesar 99,34%. Hal ini mengindikasikan bahwa faktor pengetahuan, pendidikan, dan usia memiliki pengaruh yang signifikan dalam mempengaruhi pemanfaatan Bank Sampah oleh masyarakat. Meskipun demikian, masih terdapat sebagian masyarakat yang belum atau tidak memanfaatkan Bank Sampah dengan baik. Perbedaan dengan penelitian

yang diusulkan, pada penelitian yang diusulkan lebih berfokus pendeteksian jenis sampah organik dan anorganik otomatis yang menggunakan *Machine Learning* dan, sedangkan penelitian sebelumnya lebih berfokus pada implementasi program Bank Sampah dengan algoritma Decision Tree C4.5.

Penelitian lain juga dilakukan oleh Kelvin Leonardi dkk dengan judul “Analisis Perbandingan Algoritma *Convolutional Neural Network* Dan Algoritma *Multi-Layer Perceptron Neural* Dalam Klasifikasi Citra Sampah”[4]. Penelitian ini menggunakan algoritma CNN dan MLP untuk mengklasifikasikan sampah berdasarkan gambar. Dilakukan analisis perbandingan antara kedua algoritma dalam hal kinerja klasifikasi. Metrik evaluasi yang digunakan mencakup presisi, *recall*, *f1-score*, dan akurasi. Hasil penelitian menunjukkan bahwa algoritma CNN memiliki kinerja yang lebih baik dibandingkan dengan algoritma MLP dalam mengklasifikasikan sampah. Nilai *presisi*, *recall*, *f1-score*, dan akurasi yang dihasilkan oleh algoritma CNN mencapai 0,98. Hal ini menunjukkan bahwa model CNN lebih efektif dalam membedakan jenis sampah berdasarkan kelasnya. Perbedaannya dengan penelitian yang diusulkan, penelitian yang diusulkan melibatkan penggunaan pengolahan gambar mengidentifikasi dan memilah sampah, sedangkan penelitian sebelumnya lebih berfokus pada penggunaan algoritma klasifikasi CNN dan MLP.

Yusri Yusron Nur juga melakukan penelitian dibidang IOT dengan permasalahan yang sampah yaitu pengelolaan jenis sampah dengan judul “*Matrash: The Use Of Machine Learning In The Waste Bank Based Iot Integrated With Smart Trash Bin*”. Penelitian ini bertujuan untuk merancang desain tempat sampah

otomatis menggunakan *Machine Learning* dengan memanfaatkan konsep Internet of Things (IoT) sebagai sarana untuk memilah sampah otomatis berdasarkan kategori organik dan anorganik. Desain ini menggunakan *Raspberry Pi B+* sebagai komponen utamanya. Metode pengolahan gambar digunakan untuk mengambil gambar sampah yang masuk. Tempat sampah ini dilengkapi dengan sensor ultrasonik untuk mengukur tinggi sampah dan sensor berat yang digunakan untuk menentukan harga jual sampah yang akan disetor ke bank sampah terdekat. Terdapat juga aplikasi Android yang akan memantau sampah yang masuk, sehingga membantu bank sampah dalam mengelola tempat sampah walaupun berada di luar area sampah. Perbedaan penelitian ini dengan penelitian yang diusulkan yaitu lebih berfokus penggunaan algoritma *Machine Learning* untuk mendeteksi jenis sampah berdasarkan kategori organik dan anorganik. Sementara itu, penelitian sebelumnya menggunakan *Raspberry Pi B+* sebagai komponen utama dalam desain tempat sampah otomatis, serta memanfaatkan konsep IoT.

Penelitian lain juga diusulkan oleh Sita Alden dengan judul “Implementasi Algoritma CNN Untuk Pemilahan Jenis Sampah Berbasis Android Dengan Metode CRISP-DM”. Pada penelitian ini Menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur *Mobile Net* untuk mengklasifikasikan jenis sampah berdasarkan gambar yang diambil oleh pengguna melalui aplikasi Android. Sedangkan perbedaannya dengan penelitian yang diusulkan, Penelitian yang diusulkan lebih berfokus pada pengembangan aplikasi berbasis Android dengan menggunakan algoritma *random forest* sedangkan penelitian sebelumnya

menggunakan algoritma *Convolutional Neural Network* (CNN) untuk mengklasifikasikan jenis sampah berdasarkan gambar.

B. Kajian Teori

Dalam kajian teori, peneliti mengumpulkan, meninjau, dan menganalisis literatur yang relevan yang telah diterbitkan dalam jurnal ilmiah, buku, artikel, laporan penelitian, dan sumber-sumber lain yang terpercaya

1. Pengelolaan Sampah

Pengelolaan sampah merujuk pada serangkaian kegiatan yang dilakukan untuk mengelola sampah agar dapat diolah, didaur ulang, atau dibuang dengan cara yang aman dan ramah lingkungan. Pengelolaan sampah bertujuan untuk mengurangi dampak negatif sampah terhadap kesehatan manusia, lingkungan, dan ekosistem.

Pengelolaan sampah meliputi beberapa tahapan yaitu pengumpulan, transportasi, pemilahan atau pemisahan jenis sampah, pengolahan dan pembuangan.

1. Pengumpulan: Tahap ini melibatkan kegiatan mengumpulkan semua jenis sampah dari sumbernya seperti rumah tangga, industri maupun daerah umum lainnya.
2. Transportasi: Setelah dikumpulkan di lokasi tertentu maka selanjutnya proses transportasi dimulai dengan menggunakan truk atau kendaraan lain hingga menuju tempat pembuangan akhir (TPA).

3. Pemilahan/Pemisahan: Sampah yang telah tiba di TPA kemudian dipisahkan berdasarkan jenisnya seperti organik dan anorganik serta benda-benda yang dapat didaur ulang seperti plastik dan kertas/karton.
4. Pengolahan: Pada tahap ini limbah akan diolah menjadi produk baru sehingga bisa digunakan kembali seperti proses daur ulang atau komposisi bagi limbah organik menjadi pupuk.
5. Pembuangan Terakhir :Tidak semua jenis sampah bisa didaur ulang ,sampai saat ini cara paling banyak digunakan dalam melakukan pembuangan akhir yakni dengan cara penguburan (*landfilling*) tetapi juga ada teknologi *waste to energy*(WTE)yang mampumemanfaatkan energi dari suatu bahan bakar seperti gas yang dihasilkan dari pembakaran sampah.



Gambar 2.1 Proses Pengelohan Sampah

Gambar 2.1. menjelaskan proses dari pengelolaan sampah, dimulai dari pengangkutan hingga diolah dan didaur ulang sampah-sampah yang masih layak dan yang tidak layak dibuang ke TPA. Selain langkah-langkah tersebut,

pengelolaan sampah juga melibatkan pendidikan dan kesadaran masyarakat tentang pentingnya pengelolaan sampah yang baik. Ini termasuk kampanye pengurangan sampah, edukasi tentang pemilahan sampah, dan penggalangan dukungan untuk praktik pengelolaan sampah yang berkelanjutan.

Pengelolaan sampah yang efektif dan berkelanjutan sangat penting untuk menjaga kebersihan lingkungan, mengurangi polusi, dan meminimalkan dampak negatif terhadap kesehatan manusia dan ekosistem. Pengelolaan sampah sangat penting untuk menjaga kelestarian lingkungan dan kesehatan manusia. Oleh karena itu, pemerintah dan masyarakat perlu bekerja sama dalam melakukan pengelolaan sampah dengan baik, termasuk memperkenalkan konsep 3R (*Reduce, Reuse, Recycle*) untuk mengurangi jumlah limbah yang dihasilkan[5].

2. Jenis Sampah

Jenis sampah umumnya dapat diklasifikasikan menjadi beberapa kategori berdasarkan sifat dan karakteristiknya. Berikut adalah penjelasan singkat tentang beberapa jenis sampah yang umum:

1. **Sampah Organik:** Sampah organik adalah jenis sampah yang berasal dari bahan-bahan alami yang mudah terurai, seperti sisa makanan, dedaunan, serasah, dan kertas yang terkontaminasi dengan bahan organik. Sampah organik dapat diurai oleh mikroorganisme dalam proses komposisi dan menghasilkan pupuk organik yang dapat digunakan untuk meningkatkan kesuburan tanah[6].

2. Sampah Anorganik: Sampah anorganik terdiri dari bahan-bahan yang tidak dapat terurai secara alami oleh mikroorganisme. Contoh sampah anorganik meliputi plastik, kertas dan kardus yang tidak terkontaminasi dengan bahan organik, logam, kaca, keramik, dan sebagainya. Sampah anorganik dapat didaur ulang atau diproses untuk menghasilkan bahan baku baru atau produk baru[6].
3. Sampah Berbahaya: Sampah berbahaya mengacu pada jenis sampah yang mengandung zat-zat yang dapat membahayakan kesehatan manusia dan lingkungan jika tidak dikelola dengan benar. Sampah berbahaya dapat berupa baterai bekas, limbah medis, cat, bahan kimia beracun, pestisida, obat-obatan kadaluwarsa, dan sebagainya. Sampah ini memerlukan perlakuan khusus dalam pengelolaannya, seperti pengolahan limbah medis, daur ulang baterai, atau penghapusan aman[7].
4. Sampah Elektronik (*E-Waste*): Sampah elektronik merujuk pada limbah dari peralatan elektronik seperti ponsel, komputer, televisi, kulkas, dan perangkat elektronik lainnya. *E-Waste* mengandung komponen beracun dan berbahaya seperti timbal, merkuri, kadmium, dan bahan kimia lainnya. Daur ulang dan pengolahan yang tepat diperlukan untuk mengurangi dampak negatif dari *E-Waste* terhadap lingkungan.
5. Sampah Medis: Sampah medis berasal dari fasilitas kesehatan seperti rumah sakit, klinik, atau laboratorium medis. Sampah medis termasuk jarum suntik, alat medis sekali pakai, bahan infeksius, obat-obatan kadaluwarsa, dan limbah biologis. Penanganan, pengemasan, dan pembuangan sampah

medis harus dilakukan sesuai dengan protokol keselamatan dan peraturan yang berlaku untuk mencegah penyebaran penyakit dan kerusakan lingkungan.

Pengelolaan yang tepat dari berbagai jenis sampah ini penting untuk melindungi kesehatan manusia, menjaga kebersihan lingkungan, dan meminimalkan dampak negatif terhadap alam.

3. *Machine Learning*

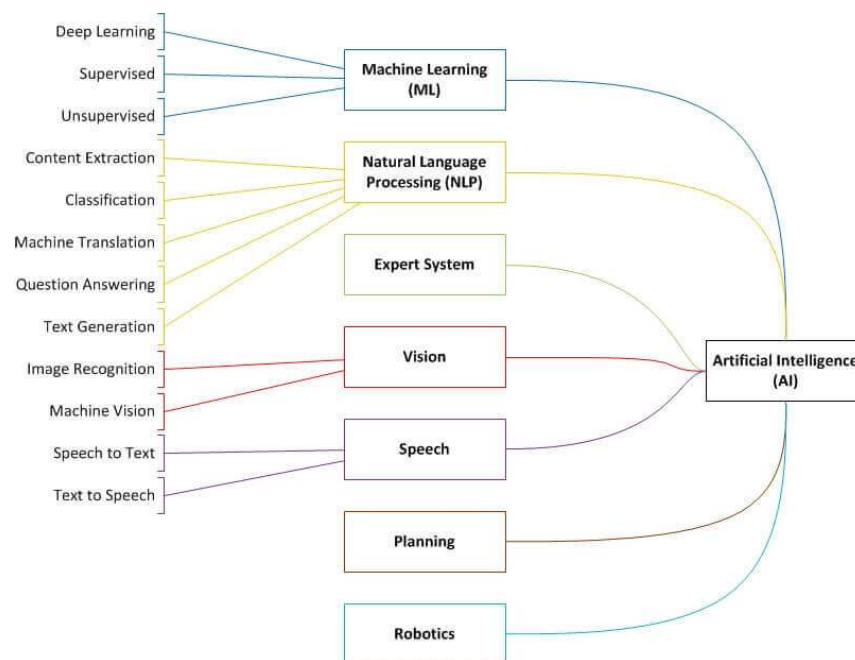
Machine learning adalah cabang dari ilmu kecerdasan buatan (AI) yang memungkinkan komputer untuk belajar dan melakukan tugas-tugas tertentu tanpa program secara eksplisit. Hal ini dilakukan dengan cara mengambil data sebagai input, menganalisisnya, kemudian memberikan keluaran/hasil berupa prediksi atau klasifikasi[7].

Dalam konteks *machine learning*, "*machine*" mengacu pada komputer atau sistem komputasi yang digunakan untuk mempelajari pola dari data, sedangkan "*learning*" mengacu pada proses di mana komputer mengubah performanya berdasarkan pengalaman yang diperoleh dari data. Dengan kata lain, komputer "belajar" dari data yang diberikan kepadanya dan menggunakan informasi tersebut untuk mengambil keputusan atau membuat prediksi di masa depan.

Proses pembelajaran mesin terjadi melalui pengenalan pola pada data input serta membuat model matematis yang dapat digunakan untuk memprediksi hasil di masa depan[8]. Model tersebut dibuat oleh algoritma yang diberi instruksi untuk mencari hubungan antara setiap variabel dalam

dataset. Dalam proses pelatihan, algoritma akan disesuaikan agar bisa mengenali pola-pola kompleks dalam data dan menjadi lebih akurat dalam membuat prediksi.

Ada beberapa jenis machine learning seperti *supervised learning*, *unsupervised learning*, *reinforcement learning* dan *deep learning*. *Supervised Learning* merupakan jenis *Machine Learning* dimana kita memberikan model sejumlah contoh “*input*” beserta contoh “*output*” yang sesuai sehingga sistem dapat belajar bagaimana cara menyelesaikan masalah baru dengan benar.



Gambar 2.2 Cabang-Cabang *Machine Learning*

Penerapan *machine learning* sangat luas dan dapat ditemukan dalam berbagai bidang, seperti pengenalan wajah, pengenalan suara, analisis data, pengambilan keputusan, pengenalan tulisan tangan, pengenalan objek, dan

masih banyak lagi. Dengan menggunakan machine learning, komputer dapat mengolah dan menganalisis data yang kompleks dengan cepat dan efisien, membantu manusia dalam membuat keputusan yang lebih baik, mengidentifikasi pola yang tersembunyi, dan meningkatkan performa sistem secara keseluruhan.

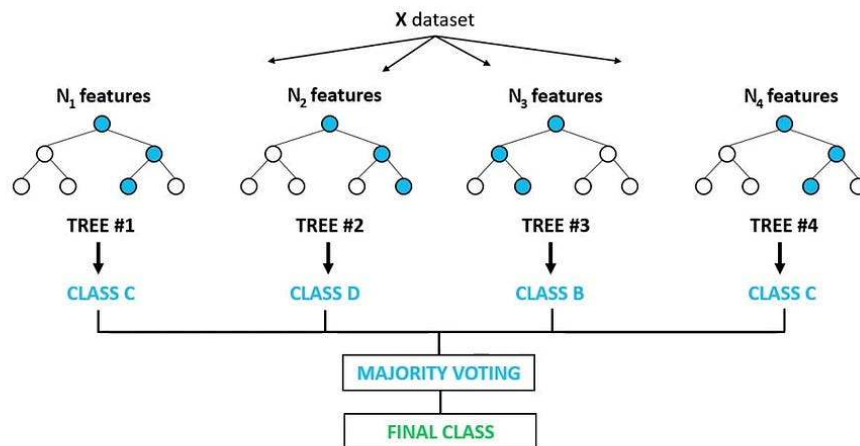
4. Algoritma *Random Forest*

Random Forest merupakan salah satu algoritma *machine learning* yang digunakan untuk tugas klasifikasi dan regresi. Algoritma ini termasuk dalam kelompok *ensemble learning*, yang menggabungkan prediksi dari beberapa pohon keputusan (*decision tree*) untuk menghasilkan hasil akhir[9].

Konsep dasar di balik *Random Forest* adalah membangun sejumlah besar pohon keputusan secara acak. Setiap pohon keputusan dalam *Random Forest* dibangun dengan menggunakan subset acak dari data pelatihan dan subset acak dari fitur yang tersedia. Dalam hal klasifikasi, setiap pohon menghasilkan prediksi kelas, dan prediksi akhir diperoleh dengan mengambil mayoritas suara dari semua pohon dalam hutan. Dalam hal regresi, prediksi akhir adalah rata-rata dari prediksi semua pohon[10].

Keunggulan utama dari *Random Forest* adalah kemampuannya untuk mengatasi *overfitting* dan menghasilkan prediksi yang lebih stabil dan akurat. Dalam proses pembangunan pohon keputusan acak, variasi yang dihasilkan oleh subset data dan fitur acak membantu mengurangi kemungkinan *overfitting*. Selain itu, *Random Forest* juga mampu menangani data yang memiliki fitur yang kurang relevan atau berkorelasi tinggi.

Random Forest Classifier



Gambar 2.3 Cara Kerja Algoritma *Random Forest*

Random Forest juga dapat memberikan informasi tentang pentingnya setiap fitur dalam tugas klasifikasi atau regresi. Dengan mengukur seberapa sering sebuah fitur digunakan untuk membagi simpul dalam pohon-pohon keputusan, kita dapat mengevaluasi tingkat kepentingan fitur tersebut dalam proses pengambilan keputusan.

Algoritma *Random Forest* telah digunakan secara luas dalam berbagai aplikasi, termasuk pengenalan pola, analisis citra, analisis teks, bioinformatika, dan banyak lagi. Karena kinerjanya yang baik dalam menghadapi data yang kompleks dan beragam, *Random Forest* menjadi salah satu pilihan populer dalam implementasi *machine learning*.

5. *Image Processing*

Image Processing atau pengolahan citra adalah cabang ilmu komputer yang berkaitan dengan manipulasi dan analisis gambar digital. Tujuannya adalah untuk memperoleh informasi yang berguna dari gambar, mengubah gambar menjadi bentuk yang lebih baik atau lebih mudah diinterpretasikan, atau melakukan tugas khusus seperti deteksi objek, pengenalan pola, restorasi gambar, atau segmentasi[11].

Proses *Image Processing* dapat dibagi menjadi beberapa tahap sebagai berikut:

1. *Preprocessing*: Tahap awal dalam proses ini melibatkan *preprocessing* gambar seperti penghapusan noise (*filtering*), peningkatan kontras serta penstabilan warna
2. Segmentasi: Tahapan selanjutnya yaitu segmentasi dimana objek di dalam gambar dipisahkan satu sama lain menjadi bagian-bagian terpisah sehingga mempermudah analisis lanjutan.
3. *Feature Extraction* : Pada tahap *feature extraction* dilakukan ekstraksi informasi penting dari setiap objek pada gambar misalnya lokalisasi garis tepi, objek geometri, dan tekstur permukaan
4. *Recognition/Classification*: Pada tahap akhir, image processing digunakan untuk melakukan klasifikasi atau identifikasi terhadap obyek yang ada pada sebuah citra dengan menggunakan algoritma *machine learning*.

Aplikasi *Image Processing* sangat luas seperti deteksi wajah, sistem keamanan digital, pengenalan karakter tulisan tangan, *optical character recognition* (OCR), diagnosis medis dan masih banyak lagi. Dalam perkembangannya, image processing memiliki peranan penting karena sebagian besar aplikasinya sudah mendukung *Artificial Intelligence* (AI) sehingga dapat memberikan prediksi hasil analisis secara otomatis tanpa adanya campur tangan manusia.

6. Bahasa Pemrograman *Python*

Python adalah bahasa pemrograman tingkat tinggi yang didesain dengan fokus pada keterbacaan kode dan kemudahan penggunaan[12]. Dibuat oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, *Python* telah menjadi salah satu bahasa pemrograman yang populer dan banyak digunakan di berbagai bidang, seperti pengembangan web, analisis data, kecerdasan buatan, pemrograman jaringan, dan lainnya.

Berikut adalah beberapa fitur dan karakteristik kunci dari *Python*:

1. **Keterbacaan Kode:** *Python* menggunakan sintaks yang bersih dan mudah dibaca, yang memungkinkan pengembang untuk mengekspresikan ide-ide kompleks dengan kode yang lebih singkat dan lebih mudah dipahami. Struktur blok kode di *Python* ditandai dengan indentasi, bukan dengan tanda kurung atau kurawal, sehingga kode cenderung terlihat lebih terstruktur.
2. **Bahasa Multiparadigma:** *Python* mendukung paradigma pemrograman berbeda, termasuk pemrograman berorientasi objek (OOP), pemrograman

prosedural, dan pemrograman fungsional. Ini memungkinkan pengembang untuk memilih pendekatan terbaik sesuai dengan kebutuhan proyek.

3. **Dukungan untuk Pustaka Ekstensif:** Python memiliki ekosistem yang kuat dengan banyak pustaka (*libraries*) dan kerangka kerja (*frameworks*) yang telah dikembangkan oleh komunitas yang aktif. Pustaka-pustaka ini memungkinkan pengembang untuk mengatasi berbagai tugas tanpa perlu menulis ulang kode dari awal, sehingga mempercepat proses pengembangan.
4. **Portabilitas:** Python adalah bahasa yang portabel, artinya kode yang ditulis dalam Python dapat dijalankan di berbagai platform dan sistem operasi tanpa perlu memodifikasi kode secara signifikan.
5. **Interpretasi dan Kompilasi:** *Python* awalnya merupakan bahasa pemrograman berbasis interpreter, yang berarti kode dieksekusi baris per baris. Namun, ada juga pustaka seperti *Cython* yang memungkinkan kode *Python* dikompilasi menjadi kode mesin, meningkatkan kinerja di beberapa kasus.
6. **Komunitas yang Besar dan Aktif:** *Python* memiliki komunitas pengembang yang besar dan aktif yang berkontribusi pada pengembangan bahasa dan pustaka-pustaka yang ada. Komunitas ini memberikan dukungan, sumber daya, dan bantuan melalui *forum*, *mailing list*, dan situs *web Python* resmi.

Python telah digunakan dalam berbagai aplikasi dan proyek terkenal, seperti Django untuk pengembangan web, *TensorFlow* untuk kecerdasan buatan, *Pandas* untuk analisis data, dan masih banyak lagi.

Contoh kode Python sederhana untuk mencetak pesan "*Hello, World!*" adalah sebagai berikut:

```
python  
  
print("Hello, World!")
```

Python adalah bahasa pemrograman yang serbaguna, ramah pengguna, dan kuat yang cocok untuk pemula maupun pengembang berpengalaman. Kemudahan penggunaan, sintaks yang bersih, serta dukungan ekosistem yang kaya membuatnya menjadi pilihan populer dalam berbagai proyek pengembangan perangkat lunak.

7. *Usecase Diagram*

Diagram Use Case adalah salah satu jenis diagram dalam pemodelan sistem yang digunakan untuk menggambarkan interaksi antara aktor (pengguna atau sistem lain) dengan sistem yang sedang dikembangkan. *Diagram Use Case* digunakan untuk memvisualisasikan fungsionalitas sistem dari perspektif pengguna. Diagram ini membantu dalam memahami kebutuhan dan perilaku yang diharapkan dari sistem.

Tabel 2.1 Diagram *Usecase*

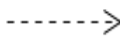
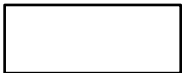
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Aktor adalah entitas eksternal yang berinteraksi dengan sistem. Aktor dapat berupa pengguna manusia, perangkat keras, sistem eksternal, atau bahkan perangkat lunak lainnya. Aktor menempati peran tertentu dalam sistem dan berhubungan dengan satu atau beberapa <i>Use Case</i> (fitur) dalam sistem.
2		<i>Include</i>	<i>Include</i> adalah hubungan antara dua <i>Use Case</i> yang menunjukkan bahwa satu <i>Use Case</i> memerlukan eksekusi dari <i>Use Case</i> lainnya. Jadi, <i>Use Case</i> yang menyertakan <i>Use Case</i> lain akan memasukkan langkah-langkah dari <i>Use Case</i> yang disertakan di dalamnya.
3		<i>Extend</i>	<i>Extend</i> adalah hubungan antara dua <i>Use Case</i> yang menunjukkan bahwa satu <i>Use Case</i> dapat memperluas fungsionalitas <i>Use Case</i> lainnya. <i>Use Case</i> ekstensi akan dilakukan hanya dalam kondisi tertentu.
4		<i>Association</i>	Garis penghubung antara aktor dan <i>Use Case</i> menunjukkan keterlibatan aktor dalam <i>Use Case</i> tertentu. Asosiasi menunjukkan bahwa aktor tertentu terlibat dalam pelaksanaan <i>Use Case</i> tersebut.
5		<i>Generalization</i>	<i>Generalization</i> adalah hubungan antara dua <i>use case</i> di mana satu <i>use case</i> adalah varian dari yang lain. <i>Use case</i> yang lebih umum akan memiliki fungsionalitas dasar, sementara <i>use case</i> yang lebih spesifik memperluas fungsionalitas tersebut dengan langkah-langkah tambahan.

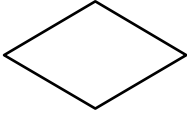
Diagram use case adalah alat yang kuat untuk menggambarkan perspektif fungsional sistem dari sudut pandang pengguna. Ini membantu tim pengembang dan *stakeholder* untuk memahami kebutuhan fungsional sistem secara jelas dan mengidentifikasi interaksi yang berbeda antara pengguna dan sistem. Dengan demikian, *diagram use case* menjadi bagian penting dari analisis kebutuhan dan perencanaan proyek perangkat lunak.

8. Flowchart

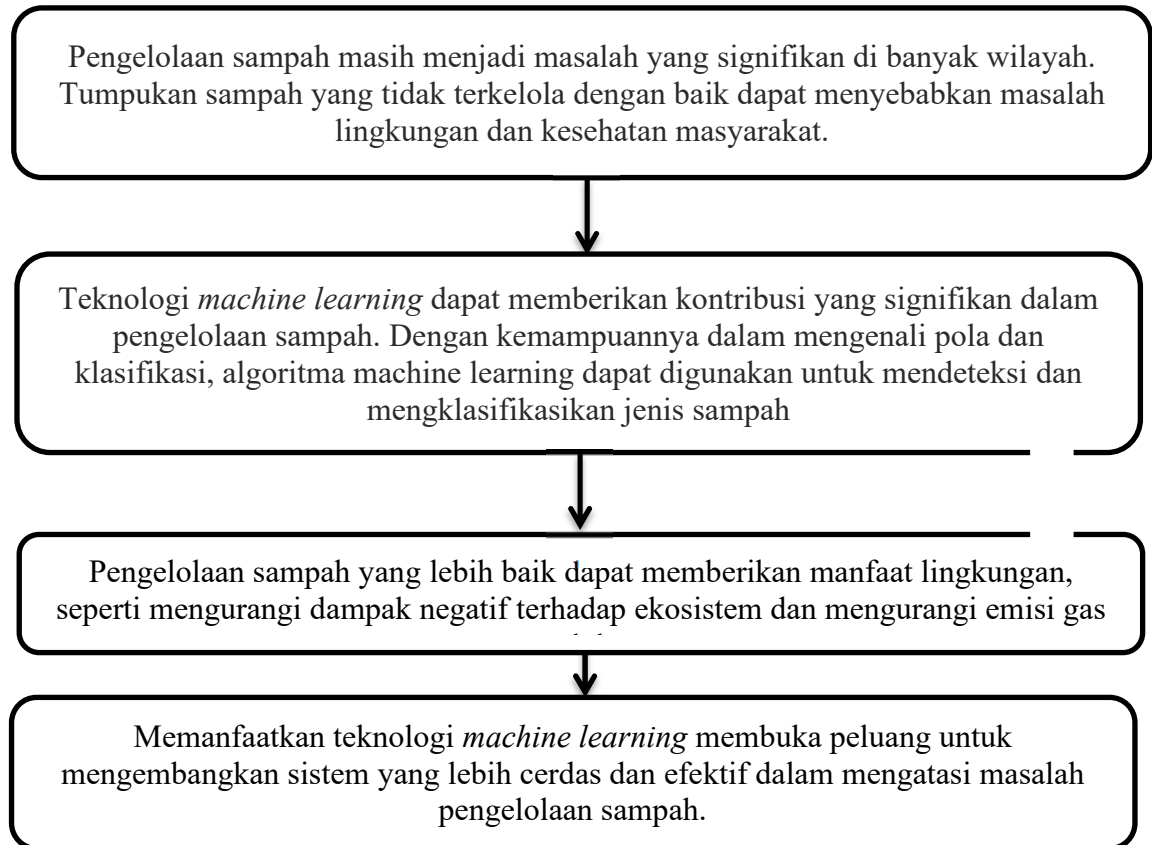
Flowchart adalah bentuk visualisasi yang menggambarkan alur logika atau proses langkah-demi-langkah dalam sebuah sistem, algoritma, atau prosedur. *Flowchart* menggunakan simbol-simbol khusus untuk mewakili langkah-langkah atau tindakan tertentu, serta penghubung untuk menunjukkan hubungan antara langkah-langkah tersebut. *Flowchart* digunakan secara luas di berbagai bidang, seperti pemrograman, bisnis, manufaktur, dan banyak lagi, untuk memahami, merancang, dan mengkomunikasikan proses dengan jelas.

Tabel 2.2 *Flowchart* dan Penjelasannya

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Start/End</i>	Simbol ini menunjukkan titik awal atau akhir dari proses atau algoritma. Digambarkan sebagai oval dengan kata " <i>Start</i> " di dalamnya untuk awal dan " <i>End</i> " untuk akhir.
2		<i>Process</i>	Simbol ini mewakili tindakan atau langkah tertentu dalam proses. Biasanya digambarkan sebagai persegi panjang dengan teks yang menjelaskan tindakan atau langkah tersebut.

NO	GAMBAR	NAMA	KETERANGAN
3		<i>Decision</i>	Simbol ini digunakan untuk mewakili kondisi atau keputusan yang mengarahkan aliran proses ke langkah berbeda, tergantung pada hasil kondisi yang diberikan. Biasanya digambarkan sebagai berlian dengan pertanyaan di dalamnya.
4		<i>Connector</i>	Simbol ini menunjukkan titik di mana aliran proses berlanjut ke halaman lain atau bagian dari <i>flowchart</i> . Ini membantu menghindari penggambaran detail yang berlebihan dan membuat <i>flowchart</i> lebih rapi.
5		<i>Input. Output</i>	Simbol ini menunjukkan proses input atau output dari atau ke dalam sistem. Misalnya, penggunaan tombol "input" atau pencetakan data sebagai "output."

C. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang dilakukan merupakan penelitian eksperimental, yaitu penelitian yang pengumpulan datanya melalui pencatatan secara langsung dari hasil percobaan yang dilakukan.

B. Alat dan Bahan Penelitian

Untuk melakukan proses penelitian dalam pembuatan aplikasi, maka diperlukan perangkat keras dan perangkat lunak guna mendukung kegiatan penelitian tersebut. Berikut ini merupakan penjelasan dari *hardware* dan *software* yang digunakan dalam pembuatan aplikasi pendeteksi jenis sampah ini.

1. Perangkat keras yang digunakan selama proses penelitian yaitu :

a. Laptop Asus ROG Strix dengan spesifikasi:

1) *Processor : Intel(r)core(tm)i7-9750h cpu@2.60hz 2.59 hzRAM
8 GB*

2) *SSD : 1 TB*

3) *Sistem Operasi : Windows 11*

b. *Smartphone oppo a16* dengan spesifikasinya :

1) *Processor : mediatek hellio g35*

2) *RAM : 3 GB*

3) *Baterai : 5000 mAh*

4) *Layar : 6,52 720x1600 hd+*

5) *Kamera Depan : 8 mp f/2.0*

6) *Kamera Belakang : 13 mp f/1.8 (wide), 8 mp f/2.2 (ultrawide), 5 mp f/2.2*

7) *Android : Android 11*

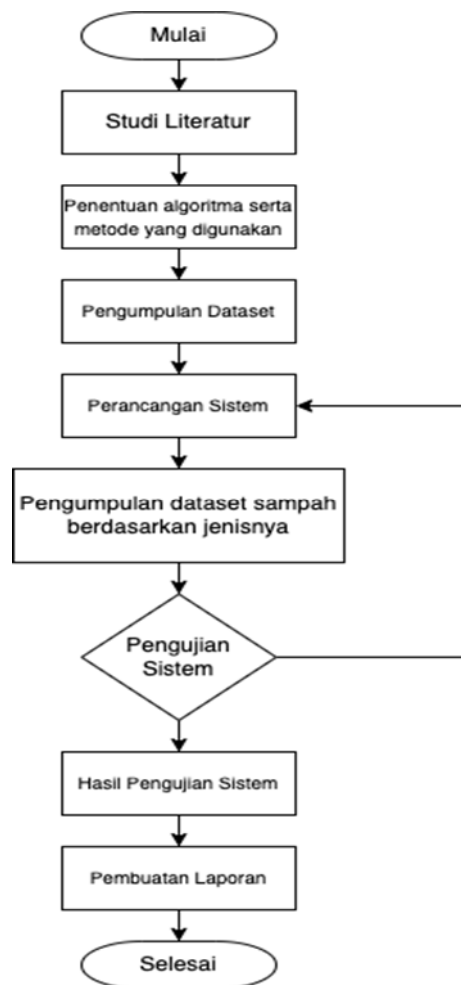
- c. Sampah sebagai dataset
- 2. *Software* yang digunakan dalam pembuatan aplikasi yaitu :
 - a. Beberapa *Library Open source* (*TensorFlow, scikit-learn, Pandas, Numpy, OpenCV* dan *Matplotlib*)
 - b. Bahasa Pemrograman *Python*
 - c. Platform *Visual Studio Code*

C. Teknik Pengumpulan Data

1. Identifikasi jenis sampah: Langkah awal adalah mengidentifikasi jenis sampah yang akan dijadikan objek penelitian. Dalam kasus ini, jenis sampah yang ingin dideteksi adalah organik dan anorganik.
2. Pemilihan lokasi
3. Penentuan jumlah sampel
4. Pengumpulan sampel, pengumpulan sampel sampah sesuai dengan jenis sampah yang ditentukan. Misalnya, untuk sampel organik, berupa sisa makanan atau bahan organik lainnya. Untuk sampel anorganik, berupa bahan-bahan seperti plastik, logam, atau kertas.
5. Pencatatan informasi
6. Pengolahan dan persiapan data

D. Tahap Penelitian

Tahap-tahap penelitian yang dimaksud dalam penelitian ini sesuai pada gambar 3.1 berikut:



Gambar 3.1 Tahapan Penelitian

2.1. Tahapan Studi Literatur

Tahapan studi literatur merupakan tahapan pencarian informasi dan referensi- referensi penelitian sebelumnya yang terkait dengan informasi system pendeteksi jenis sampah, metode pengolahan citra secara umum dan algoritma yang digunakan untuk klasifikasi.

2.2.Tahapan Analisis dan Desain Sistem

Pada tahap analisis dilakukan perumusan masalah, batasan-batasan masalah dan bagaimana menyelesaikan masalah tersebut. Permasalahan yang diangkat pada penelitian ini adalah bagaimana system ini mampu mendeteksi jenis sampah berdasarkan karakteristik yang telah ditentukan pada ekstraksi fitur.

2.3.Tahapan Implementasi.

Tahapan implementasi adalah tahapan penerapan tahapan desain system dengan harapan tahapan ini menghasilkan keluaran yang sesuai dengan tahap analisis.

2.4.Tahapan Uji Coba dan Analisis Hasil

Tahapan uji coba dan analisa hasil merupakan tahapan dimana sistem yang telah diimplementasikan sesuai dengan desain sistem akan diuji coba pada tahapan ini. Uji coba dilakukam sesuai dengan skenario uji coba yang telah dirancang pada tahap desain sistem. Berdasarkan hasil uji coba tersebut, sistem akan dianalisis untuk mengetahui performa dari sistem yang dibangun.

2.5.Tahapan Penulisan Laporan

Tahapan ini merupakan tahapan penulisan laporan dari keseluruhan tahapan yang dilakukan pada penelitian ini. Setiap tahapan akan didokumentasikan secara tertulis sebagai bukti dari hasil penelitian yang telah dilakukan.

E. Metode Pengujian

Metode pengujian ini menggunakan :

1. Persiapan Data Pengujian: Persiapan dataset pengujian yang terpisah dari data pelatihan.
2. Pengujian Model: Menggunakan model *machine learning* yang telah dilatih pada data pelatihan untuk memprediksi jenis sampah pada dataset pengujian. Data sampah ke dalam model dan peroleh prediksi untuk setiap sampah.
3. Evaluasi Hasil: Membandingkan prediksi model dengan label sebenarnya pada dataset pengujian. Hitung metrik evaluasi seperti akurasi, presisi, recall, dan *F1-score* untuk mengevaluasi kinerja model dalam mendeteksi jenis sampah dengan benar.
4. Analisis Kesalahan: Analisis kesalahan yang terjadi pada model. Identifikasi jenis sampah yang sering salah diprediksi dan cari tahu faktor-faktor yang mempengaruhi kesalahan tersebut. Hal ini dapat membantu dalam memahami kelemahan model dan mengidentifikasi area yang perlu diperbaiki.
5. Validasi Silang (*Cross-Validation*): Jika dataset yang digunakan terbatas, lakukan validasi silang untuk menguji kinerja model secara lebih akurat. Dalam validasi silang, dataset dibagi menjadi beberapa bagian (*fold*), lalu dilakukan pelatihan dan pengujian model secara berulang dengan menggunakan *fold* yang berbeda sebagai data pelatihan dan pengujian.
6. Uji Coba dengan Data Baru: Setelah model dinilai melalui pengujian internal, uji coba model dengan menguji performanya pada dataset sampah

yang belum pernah dilihat sebelumnya. Hal ini dapat memberikan gambaran tentang kehandalan model saat diterapkan pada situasi nyata.

7. Kesimpulan: Berdasarkan hasil pengujian, kesimpulan tentang keefektifan dan keandalan model dalam mendeteksi jenis sampah. Identifikasi kelebihan dan kelemahan model serta saran untuk perbaikan atau pengembangan lebih lanjut.

F. Jenis Data

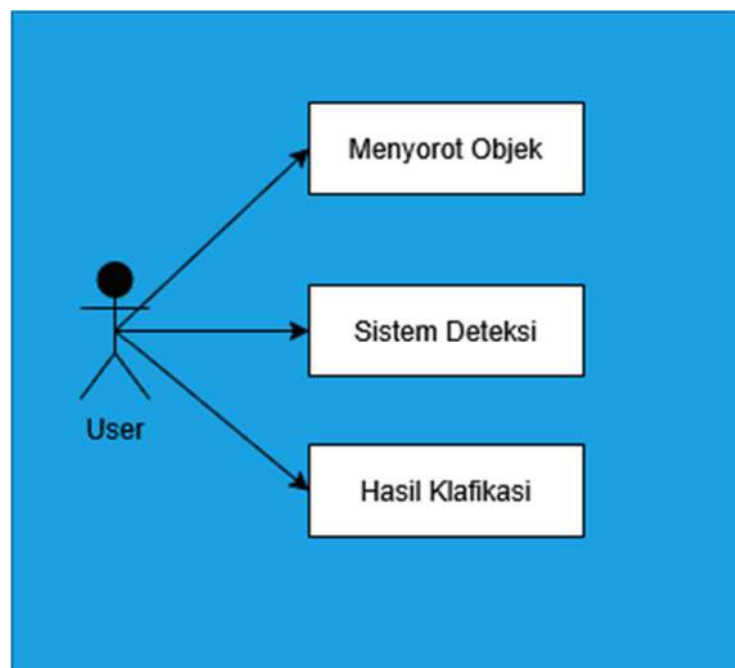
Jenis data yang dikumpulkan dalam penelitian ini adalah sebagai berikut:

1. Data Sampah: Data sampah menjadi komponen utama dalam penelitian ini. Data ini dapat berupa gambar/gambar sampah yang telah dikumpulkan dari berbagai sumber atau sumber data lainnya yang relevan. Setiap data sampah harus memiliki label yang menunjukkan apakah itu merupakan sampah organik atau anorganik.
2. Data Latih: Data latih digunakan untuk melatih model *machine learning*. Data ini terdiri dari kumpulan sampah yang telah diklasifikasikan sebelumnya sebagai organik atau anorganik. Data pelatihan harus mencakup representasi yang seimbang dari kedua jenis sampah untuk mencapai hasil yang akurat.
3. Data Uji: Data uji digunakan untuk menguji kinerja model yang telah dilatih. Data ini juga harus mencakup sampel yang mewakili baik sampah organik maupun anorganik. Data pengujian harus terpisah secara independen dari data pelatihan.

4. Data Validasi: Data validasi digunakan untuk melakukan validasi model selama proses pengembangan. Data ini membantu dalam mengoptimalkan parameter model dan menghindari overfitting.

G. Desain Sistem

1. Diagram Usecase



Gambar 3.2 Use Case Diagram

Penjelasan :

1. *User* adalah individu atau entitas yang menggunakan sistem untuk mendeteksi jenis sampah. Mereka berinteraksi dengan sistem melalui antarmuka pengguna.
2. Menyorot Objek: *User* menyorot objek sampah menggunakan kamera ponsel atau perangkat lainnya. Gambar sampah ini akan menjadi input untuk proses deteksi dan klasifikasi.

3. Sistem Deteksi: Sistem menerima gambar sampah dan menggunakan algoritma klasifikasi (misalnya *Random Forest*) untuk mendeteksi jenis sampah. Sistem menganalisis fitur-fitur dari gambar dan mengklasifikasikan sampah sebagai organik atau anorganik.
4. Menampilkan Hasil Klasifikasi: Setelah klasifikasi selesai, hasilnya ditampilkan kepada pengguna. Informasi tentang jenis sampah (organik atau anorganik) akan ditampilkan melalui antarmuka pengguna, misalnya dalam bentuk teks atau ikon.
5. Programmer: Peran programmer dalam pengembangan sistem deteksi sampah adalah untuk mengembangkan algoritma dan membangun model klasifikasi.
6. Mengembangkan Algoritma: Programmer mengembangkan algoritma yang digunakan dalam proses klasifikasi sampah. Mereka memilih dan menerapkan algoritma *machine learning* yang tepat, seperti Random Forest, dan melakukan pengaturan parameter yang diperlukan.
7. Membangun Model Klasifikasi: Programmer membangun model klasifikasi menggunakan algoritma yang telah dikembangkan. Dengan cara mengumpulkan dan mempersiapkan data pelatihan yang mencakup berbagai jenis sampah, melatih model dengan menggunakan data tersebut, dan mengoptimalkan performa model.
8. Memvalidasi dan Menguji Model: Programmer melakukan validasi dan pengujian model klasifikasi untuk memastikan bahwa model dapat mengklasifikasikan jenis sampah dengan akurasi tinggi.

9. Menyediakan *User Interface*: Programmer menyediakan *User Interface* yang memungkinkan pengguna untuk berinteraksi dengan sistem deteksi sampah. Antarmuka pengguna dapat berupa aplikasi mobile, aplikasi web, atau antarmuka lainnya yang memungkinkan pengguna untuk mengambil gambar sampah, mengirimkannya ke sistem, dan melihat hasil klasifikasi.

H. Desain Interface

1. Tampilan Menu Utama

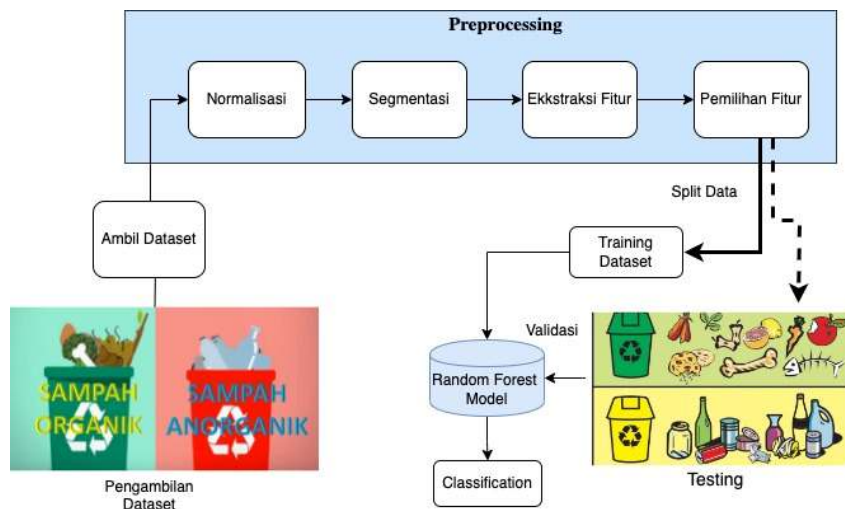


Gambar 3.3 Tampilan Menu Utama

- a) Pengambilan Gambar: Fitur ini memungkinkan pengguna untuk mengambil gambar sampah dengan menggunakan kamera perangkat. Terdapat tombol atau ikon yang memicu kamera untuk mengambil gambar sampah.
- b) Input Gambar : Fitur ini memberikan pengguna agar menggunakan gambar yang telah disediakan sebagai data uji.
- c) Hasil Klasifikasi: Setelah proses klasifikasi selesai, halaman ini menampilkan hasil klasifikasi jenis sampah berdasarkan gambar yang diambil. Informasi tentang jenis sampah, seperti organik atau anorganik, ditampilkan dengan jelas.

- d) Riwayat atau Log: Fitur ini memungkinkan pengguna untuk melihat riwayat atau log dari gambar sampah yang telah diklasifikasikan sebelumnya. Pengguna dapat melihat daftar gambar yang diklasifikasikan beserta tanggal dan waktu klasifikasinya.
- e) Pengaturan: Halaman ini berisi pengaturan aplikasi atau sistem, seperti pengaturan notifikasi, preferensi pengguna, atau opsi lain yang dapat disesuaikan.
- f) *LogOut* : Tombol yang berfungsi untuk keluar dari aplikasi

2. Rancangan Sistem



Gambar 3.4 Rancangan Sistem Aplikasi

Penjelasan :

1. Dataset: Langkah ini melibatkan pengambilan gambar sampah menggunakan kamera atau pilihan pengunggahan gambar dari perangkat pengguna.

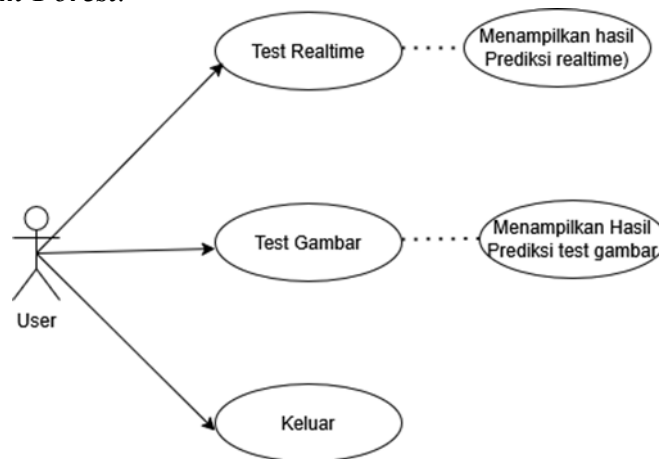
2. Preprocessing: Langkah ini mencakup pemrosesan gambar yang diambil untuk mempersiapkannya sebelum dilakukan pengolahan lebih lanjut. Ini mungkin termasuk penyesuaian ukuran, normalisasi, atau peningkatan kontras gambar.
3. Ekstraksi Fitur: Langkah ini melibatkan ekstraksi fitur dari gambar sampah yang telah diproses. Ini dapat dilakukan menggunakan berbagai metode, seperti ekstraksi fitur menggunakan algoritma HoG atau ekstraksi fitur menggunakan *Convolutional Neural Network* (CNN).
4. Modelling: Setelah fitur-fitur diekstraksi, langkah selanjutnya adalah melakukan klasifikasi jenis sampah. Metode klasifikasi yang digunakan, seperti *Random Forest*, akan memproses fitur-fitur tersebut dan memberikan prediksi tentang jenis sampah.
5. Klasifikasi: Langkah terakhir dalam sistem adalah menampilkan hasil klasifikasi kepada pengguna. Ini dapat berupa informasi tentang jenis sampah, persentase kepercayaan, atau tampilan visual lainnya yang relevan.
6. Selesai: Langkah akhir dalam flowchart adalah mengakhiri sistem deteksi sampah.

BAB IV

HASIL DAN PEMBAHASAN

A. Use Case Diagram Sistem Deteksi Sampah

Use Case Diagram ini menggambarkan interaksi antara dua aktor utama dalam sistem, yaitu **User** (Pengguna) dan **Programmer** (Pengembang Sistem), serta bagaimana mereka berinteraksi dengan aplikasi untuk mengklasifikasikan sampah organik dan anorganik menggunakan algoritma **Random Forest**.



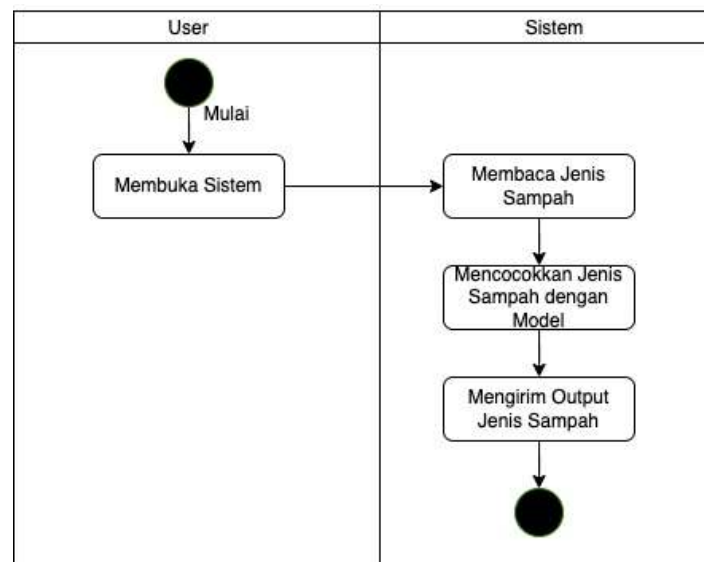
Gambar 4.1 *Use Case Diagram* Deteksi Sampah

Tabel 4 1 Deskripsi Use Case Deteksi Sampah

Aktor	Use Case	Deskripsi
User	Login	Pengguna melakukan login ke dalam <i>system</i> untuk mengakses fitur-fitur aplikasi
User	Test Realtime	Pengguna menggunakan fitur untuk mendeteksi jenis sampah secara real-time melalui kamera.
User	Test Gambar	Pengguna mengunggah gambar sampah untuk diklasifikasikan oleh sistem.
User	Keluar	Pengguna melakukan logout dari sistem.

B. Activity Diagram Sistem Deteksi Sampah

Activity Diagram ini menggambarkan alur aktivitas yang terjadi dalam sistem ketika pengguna ingin mengklasifikasikan gambar sampah (organik atau anorganik) menggunakan aplikasi berbasis algoritma **Random Forest**. Diagram ini akan menunjukkan langkah-langkah yang diambil mulai dari pengguna mengunggah gambar hingga sistem menampilkan hasil klasifikasi.

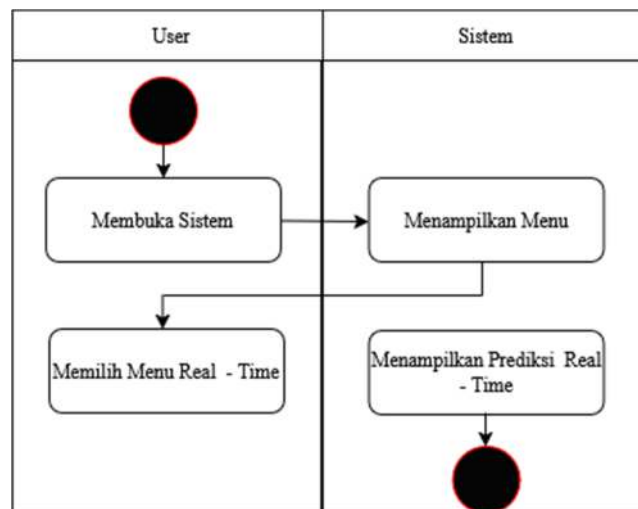


Gambar 4.2 Activity Diagram Sistem Deteksi Sampah

Penjelasan :

Aktivitas dimulai dengan pengguna membuka sistem kemudian memilih gambar untuk diuji. Gambar yang diunggah melalui aplikasi akan diproses terlebih dahulu oleh sistem untuk dibaca jenis sampahnya dan dicocokkan dengan model *random forest* yang telah dibuat. Gambar yang telah diproses dan fitur yang diekstraksi kemudian dimasukkan ke dalam model *Random Forest* untuk dilakukan

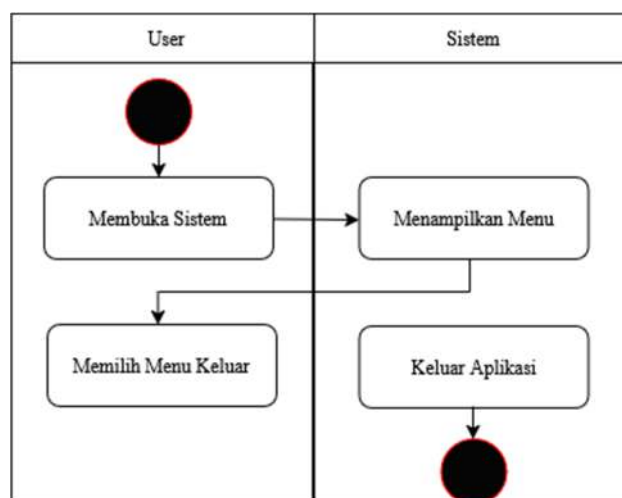
klasifikasi. Model akan memutuskan apakah gambar tersebut termasuk **Sampah Organik** atau **Sampah Anorganik**.



Gambar 4.3 Activity Diagram Realtime

Penjelasan:

Aktivitas dimulai dengan pengguna membuka sistem kemudian memilih menu realtime. Gambar yang diunggah melalui aplikasi akan diproses terlebih dahulu oleh sistem untuk dibaca jenis sampahnya dan dicocokkan dengan model random forest yang telah dibuat. Model akan memutuskan apakah gambar tersebut termasuk **Sampah Organik** atau **Sampah Anorganik**.



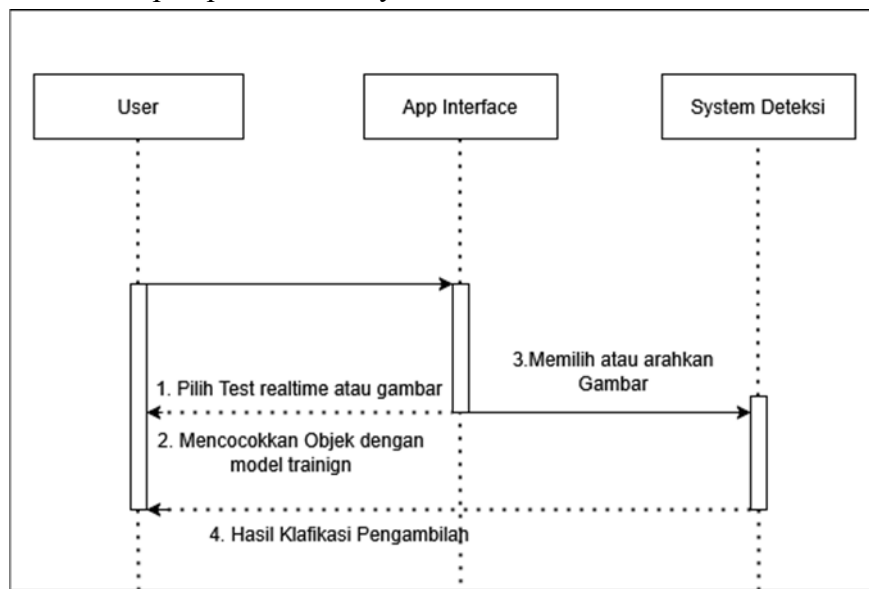
Gambar 4.4 Activity Diagram Realtime

Penjelasan :

Aktivitas dimulai dengan pengguna membuka sistem kemudian memilih menu keluar. *System* Akan Keluar dari aplikasi

C. *Sequence Diagram* Sistem Deteksi Sampah

Sequence Diagram merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan *message* (pesan) apa yang dikirim dan kapan pelaksanaannya.

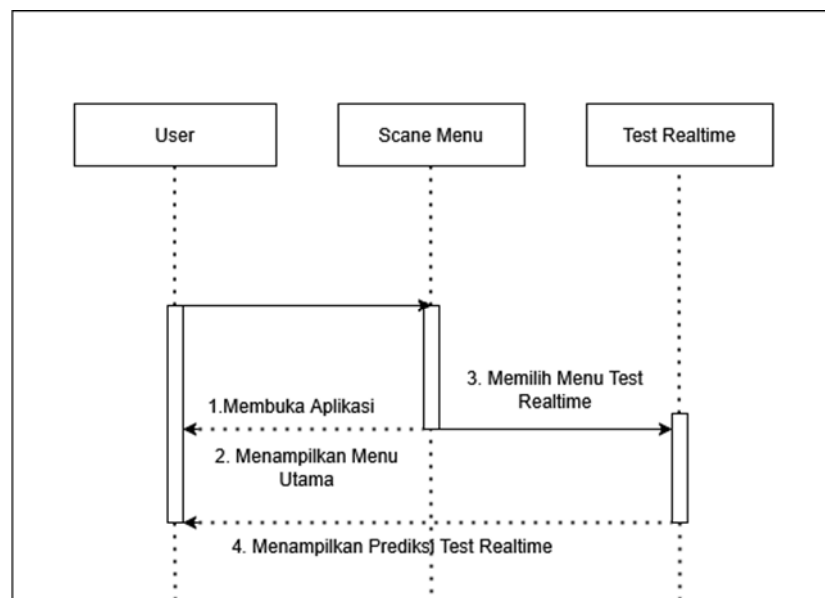


Gambar 4.5 Diagram *Sequence* Sistem Deteksi Sampah

Penjelasan :

- a. *User* (Pengguna): Aktor yang menggunakan aplikasi untuk mendeteksi jenis sampah, baik melalui gambar yang diunggah atau melalui kamera *real-time*.
- b. *App Interface* (Antarmuka Aplikasi): Komponen aplikasi yang menjadi perantara antara *user* dan sistem. Di sini, *user* melakukan interaksi seperti memilih opsi dan melihat hasil klasifikasi.

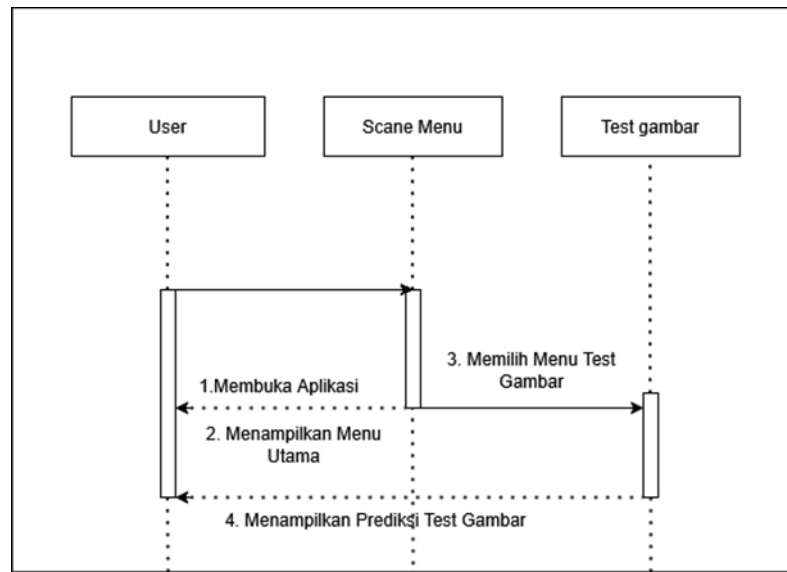
- c. *System* deteksi : Model *Random Forest* yang bertanggung jawab untuk memproses gambar dan mengklasifikasikan jenis sampah menjadi organik atau anorganik.



Gambar 4.6 Diagram *Sequence realtime*

Penjelasan :

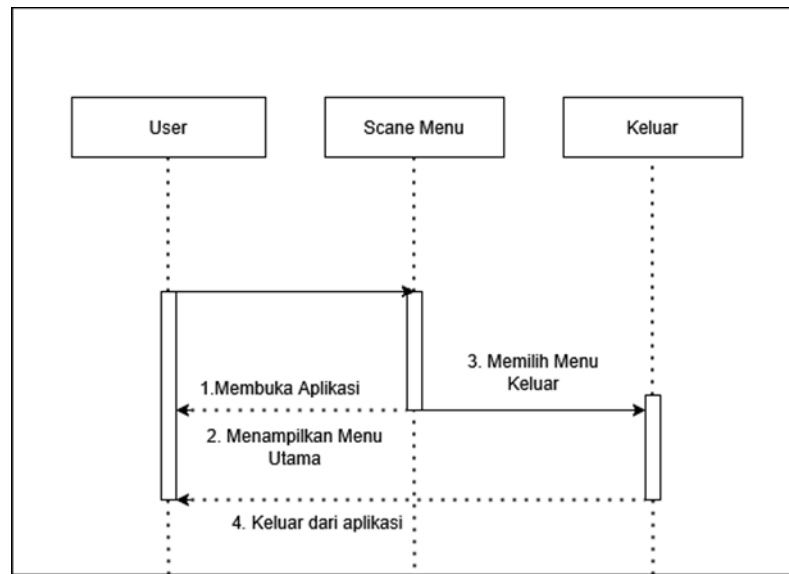
- a. *User* (Pengguna): Aktor yang menggunakan aplikasi untuk mendeteksi jenis sampah, melalui kamera *real-time*.
- b. (*Scane Menu*): Komponen aplikasi yang menjadi perantara antara user dan sistem. Di sini, user melakukan interaksi seperti memilih menu.
- c. *Realtime*: bertanggung jawab untuk memproses hasil prediksi secara langsung dan mengklasifikasikan jenis sampah menjadi organik atau anorganik.



Gambar 4.7 Diagram *Sequence test gambar*

Penjelasan :

- A. *User* (Pengguna): Aktor yang menggunakan aplikasi untuk mendeteksi jenis sampah, melalui gambar yang diunggah.
- B. (*Scane Menu*): Komponen aplikasi yang menjadi perantara antara *user* dan sistem. Di sini, user melakukan interaksi seperti memilih menu..
- C. *System* deteksi Test gambar: Model *Random Forest* yang bertanggung jawab untuk memproses gambar dan mengklasifikasikan jenis sampah menjadi organik atau anorganik.



Gambar 4.8 Diagram *Sequence test* gambar

Penjelasan :

- A. *User* (Pengguna): Aktor yang menggunakan aplikasi untuk mendeteksi jenis sampah, melalui gambar yang diunggah.
- B. (*Scane Menu*): Komponen aplikasi yang menjadi perantara antara *user* dan sistem. Di sini, *user* melakukan interaksi seperti memilih keluar dari aplikasi
- C. *System Keluar* : System yang mengatur keluarnya aplikasi.

D. Deskripsi Dataset

Dataset yang digunakan dalam penelitian ini bersumber dari hasil pengumpulan data gambar berbagai jenis sampah organik dan anorganik. Adapun rincian dataset adalah sebagai berikut:

1. Kategori Sampah

Sampah dalam dataset dikategorikan menjadi dua kelompok utama berdasarkan sifat materialnya, yaitu organik dan anorganik. Sampah organik adalah sampah yang berasal dari bahan-bahan alami yang dapat terurai, seperti sisa makanan, dedaunan, dan limbah dapur. Proses dekomposisi sampah organik terjadi melalui bantuan mikroorganisme, sehingga jenis sampah ini dapat dimanfaatkan kembali sebagai kompos atau pupuk organik. Di sisi lain, sampah anorganik terdiri dari bahan-bahan sintetis atau hasil olahan manusia yang sulit terurai secara alami, seperti plastik, logam, dan kaca. Sampah anorganik umumnya membutuhkan proses daur ulang yang melibatkan teknologi dan biaya tambahan untuk diolah kembali menjadi barang yang dapat digunakan. Pengelompokan ini bertujuan untuk mempermudah pengelolaan dan pengolahan sampah sesuai dengan karakteristik masing-masing kategori. Berikut adalah contoh beberapa gambar sampah yang menjadi dataset penelitian.

Tabel 4.2 Tabel Dataset Gambar

Kategori	Jenis Sampah	Contoh Gambar
SAMPAH ORGANIK	Sisa Nasi	

Kategori	Jenis Sampah	Contoh Gambar
	Kulit Telur	
	Kulit Pisang	
	Jeruk Nipis	
	Cabai Rawit	
	Kertas	
	Daun	
	Roti	
	Bawang Merah	

Kategori	Jenis Sampah	Contoh Gambar
ANORGANIK	Batang Kayu	
	Botol Minuman	
	Gelas Plastik	
	Pembungkus Deterjen	
	Pembungkus Pewangi	
	Pembungkus Snack	
	Kantong Plastik	

Kategori	Jenis Sampah	Contoh Gambar
	Kaleng Minuman	
	Ban Bekas	
	Botol Kaca	
	Kabel	

2. Jumlah Data:

- Setiap jenis sampah organik memiliki **250 sampel gambar** dan setiap sampah anorganik memiliki **125 sampel gambar**.
- Total keseluruhan dataset adalah **2.000 sampel gambar**, yang terdiri dari **1000 sampel untuk sampah organik** dan **1000 sampel untuk sampah anorganik**.
- Pembagian Dataset:** Data yang digunakan dibagi menjadi dua bagian, yakni **data latih** dan **data uji** dengan pembagian **80% untuk data latih** dan **20% untuk data uji**. Pembagian ini bertujuan

untuk memastikan model memiliki cukup data untuk belajar dan diuji secara akurat dalam mengklasifikasikan sampah.

3. Proses *Preprocessing* Data:

- a. **Resizing Gambar:** Gambar diubah ukurannya agar seragam sesuai dengan kebutuhan model *machine learning*. Fungsi dari *resizing* adalah untuk memastikan konsistensi dimensi input sehingga model dapat memproses data dengan efisien.
- b. **Augmentasi Data:** Data diperluas dengan melakukan transformasi seperti rotasi, *flipping*, atau penambahan *noise*. Tujuannya adalah meningkatkan keragaman dataset sehingga model dapat lebih robust terhadap variasi data di dunia nyata.
- c. **Normalisasi:** Data dinormalisasi agar nilai piksel berada dalam rentang tertentu (misalnya, 0-1). Normalisasi bertujuan untuk mempercepat proses pelatihan dan meningkatkan akurasi model.

4. Fitur Utama:

- a. Fitur yang diambil dari setiap objek sampah adalah:
 - **Warna:** Berfungsi untuk membedakan jenis sampah berdasarkan karakteristik visualnya.
 - **Bentuk:** Membantu mengenali kontur dan pola yang khas dari tiap objek.
 - **Tekstur:** Digunakan untuk mengidentifikasi pola permukaan, misalnya tekstur halus pada plastik atau kasar pada kulit telur.

E. Implementasi Model *Machine Learning*

1. Algoritma Klasifikasi yang Digunakan

Algoritma klasifikasi yang diterapkan dalam penelitian ini adalah *Random Forest*. *Random Forest* adalah metode ensemble learning yang menggabungkan beberapa pohon keputusan (*decision trees*) untuk meningkatkan akurasi dan kestabilan prediksi. Keunggulan utama algoritma ini adalah kemampuannya dalam menangani data yang tidak seimbang dan memberikan informasi tentang pentingnya setiap fitur dalam proses klasifikasi.

2. Parameter Model yang Digunakan

Pembuatan model dilakukan menggunakan beberapa langkah dan parameter penting yang dirancang untuk memastikan efisiensi dan akurasi. Berikut ini adalah rincian proses dan parameter yang digunakan:

- a. **Ekstraksi Histogram Warna:** Histogram warna dalam ruang HSV diekstraksi dari setiap gambar menggunakan 32 bins untuk masing-masing saluran (*Hue*, *Saturation*, dan *Value*). Histogram dinormalisasi dan digabungkan menjadi satu vektor fitur.
- b. **Ukuran Gambar:** Setiap gambar diubah ukurannya menjadi 64x64 piksel untuk konsistensi input.
- c. **Parameter Random Forest:**
 - a. $n_estimators = 100$: Jumlah pohon keputusan yang digunakan.
 - b. $random_state = 42$: Menjamin hasil yang konsisten antara pelatihan ulang.
 - c. Fitur histogram warna sebagai input utama untuk pelatihan.

3. Alur Implementasi

Implementasi model dilakukan melalui tahapan berikut:

a. Input Data:

- Data sampah yang telah melalui proses *preprocessing* (*resizing*, *augmentasi*, dan *normalisasi*) digunakan sebagai input.
- Dataset dibagi menjadi dua bagian: 80% untuk data latih dan 20% untuk data uji.

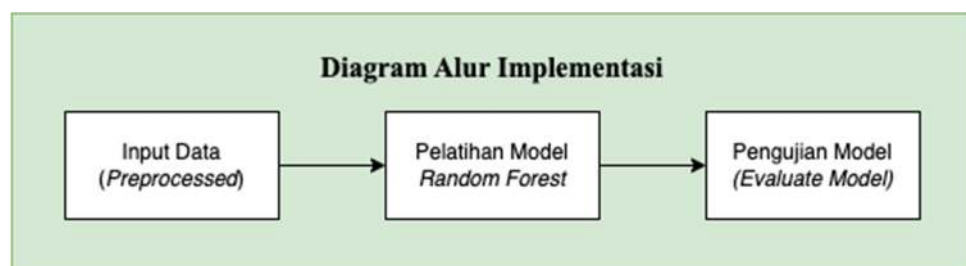
b. Pelatihan Model:

1. Data latih digunakan untuk melatih model *Random Forest*. Algoritma akan membangun sejumlah pohon keputusan berdasarkan subset acak dari data.
2. Pada setiap pohon, algoritma secara acak memilih subset fitur untuk menentukan pemisahan terbaik pada simpulnya.

c. Pengujian Model:

- Model yang telah dilatih diuji menggunakan data uji.
- Prediksi model dibandingkan dengan label sebenarnya untuk menghitung akurasi, presisi, *recall*, dan *F1-score*.

4. Diagram Alur Implementasi



Gambar 4.9 Diagram Alur Implementasi

Tahapan ini memastikan bahwa model tidak hanya belajar dari data yang tersedia, tetapi juga mampu memberikan prediksi yang akurat terhadap data baru yang belum pernah dilihat sebelumnya. *Model Random Forest* dipilih karena keandalannya dalam mengelola data dengan karakteristik yang beragam serta kemampuannya untuk memberikan hasil yang stabil dan dapat diinterpretasikan.

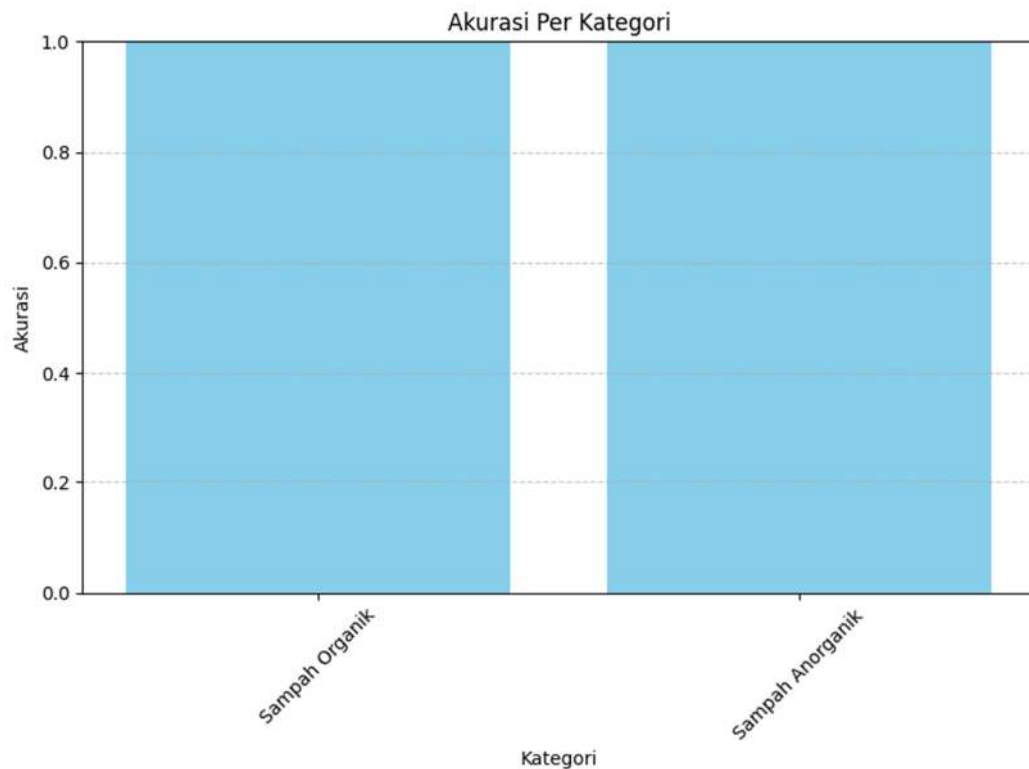
F. Hasil Eksperimen

1. Akurasi Model

Hasil eksperimen menunjukkan performa model *Random Forest* dalam mengklasifikasikan data sampah berdasarkan fitur histogram warna. Akurasi model dihitung untuk data training dan data testing, dengan hasil sebagai berikut:

- **Data Training:** Akurasi mencapai 100%, menunjukkan bahwa model mampu mempelajari pola dari data training dengan baik.
- **Data Testing:** Akurasi sebesar 100%, yang menunjukkan bahwa model dapat melakukan generalisasi dengan cukup baik terhadap data yang belum pernah dilihat sebelumnya.

Persamaan antara akurasi data training dan data testing menunjukkan bahwa model tidak mengalami *overfitting*. Selain itu, hasil akurasi untuk setiap kategori juga dihitung dan divisualisasikan menggunakan diagram batang. Diagram ini membantu dalam mempermudah pemahaman terkait tingkat akurasi model untuk masing-masing kelas sampah. Gambar berikut menunjukkan akurasi model untuk setiap kategori.



Gambar 4.10 Akurasi Pembuatan Model Menggunakan *Random Forest*

Dari diagram tersebut, dapat terlihat bahwa model menunjukkan akurasi yang sangat baik pada kategori sampah organik dan anorganik, dengan akurasi sebesar 100%. Hal ini menunjukkan bahwa model dapat mengenali karakteristik setiap jenis sampah dengan sangat baik berdasarkan analisis histogram warna.

2. *Confusion Matrix*

Confusion matrix digunakan untuk mengevaluasi performa model dalam mengklasifikasikan data pada setiap kategori. Matriks ini memperlihatkan jumlah prediksi yang benar dan salah yang dilakukan oleh model pada masing-masing kelas. Pada gambar berikut, kita dapat melihat matriks kebingungan untuk model yang telah diuji sebanyak 10 kali, dengan

dua kelas yaitu *Sampah Organik* dan *Sampah Anorganik*. Berikut gambar *confusion matrix* terdiri dari empat metrik utama, yaitu *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN). Setiap metrik ini merepresentasikan jumlah data yang diklasifikasikan dengan benar atau salah oleh model, tergantung pada kelas target dan kelas prediksi yang terkait.

		AKTUAL	
		Positif	Negatif
PREDIKSI	Positif	<i>True Positive</i> (TP)	<i>False Positive</i> (FP)
	Negatif	<i>False Negative</i> (FN)	<i>True Negative</i> (TN)

Gambar 4.11 Parameter Utama dari *Confusion Matrix*

***True Positive* (TP)** : Jika prediksi menyatakan benar maka hasil nya juga menyatakan benar

***True Negative* (TN)** : jika prediksi menyatakan salah maka hasilnya memang salah

***False Positive* (FP)** : : Jika prediksi menyatakan benar tetapi hasilnya salah

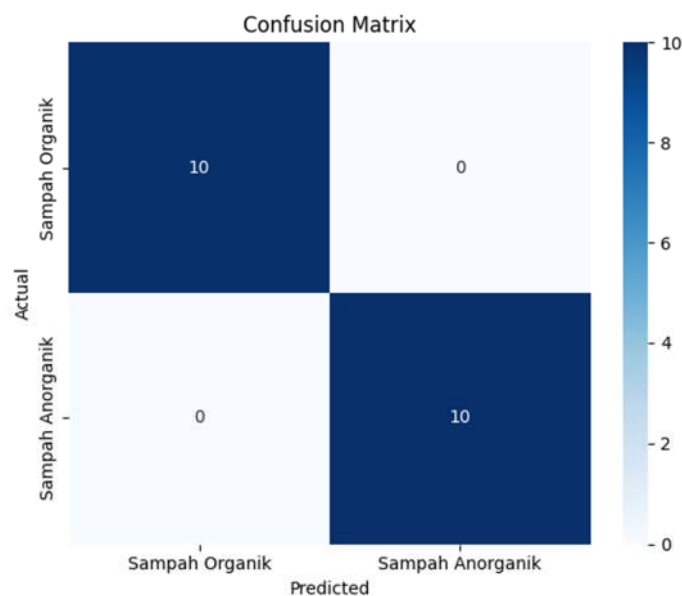
***False Negative* (FN)** : Jika Prediksi menyatakan salah dan hasilnya Benar

Setelah mendapatkan *confusion matriks* dari sebuah model maka beberapa nilai variabel dapat dicari berdasarkan data dari *confusion matriks*

tersebut. Salah satu variabel itu adalah menghitung akurasi. Data pada sistem pendeteksi sampah sangat cocok untuk menghitung akurasi karena jenis tugas yang dilakukan oleh sistem ini adalah klasifikasi, di mana setiap jenis sampah mewakili satu kelas. Oleh karena itu, untuk mengukur kinerja model pendeteksi sampah ini, metrik evaluasi yang umum digunakan adalah akurasi. Akurasi adalah rasio hasil prediksi yang benar (positif dan negatif) dengan jumlah keseluruhan data.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \dots\dots\dots (1)$$

Pada matriks ini, setiap baris mewakili kelas aktual (sesuai dengan data uji), sementara setiap kolom mewakili kelas prediksi yang dihasilkan oleh model. Berikut adalah interpretasi dari confusion matrix yang ditampilkan:



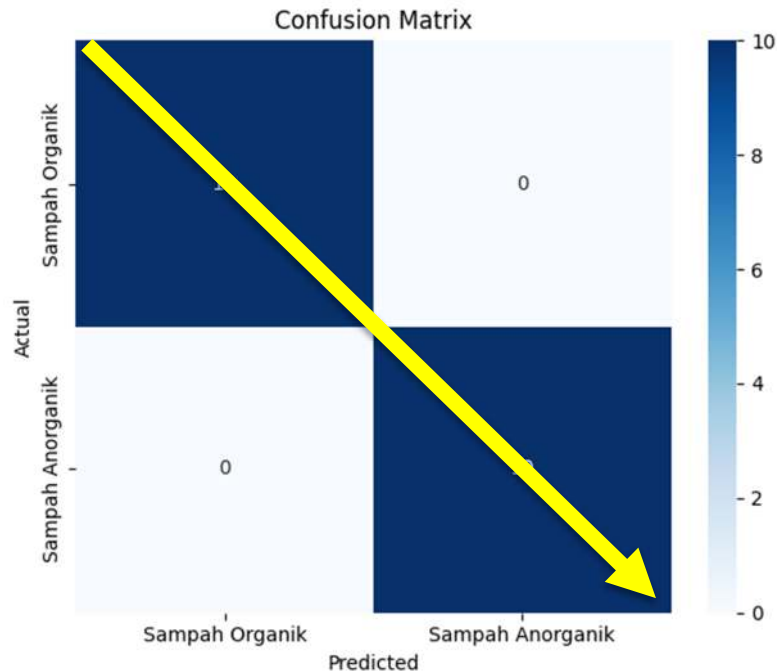
Gambar 4.12 *Confusion Matrix* dari Pembuatan Model

Confusion matriks dibuat dari model yang terpilih berdasarkan akurasi yang tertinggi agar proses validasi dan evaluasi bisa dilakukan terhadap model yang dibangun. Selain itu, hasil confusion matriks yang didapatkan bisa dibandingkan dengan hasil yang di dapatkan pada pengukuran atau percobaan langsung terhadap objek yang diteliti.

Untuk menghitung nilai akurasi maka terlebih dahulu harus dicari nilai TP, TN, FP dan FN pada matriks yang *multiclass* ini. Berikut penjelasan bagaimana mencari nilai variabel dari confusion matriks :

a. Menghitung Nilai TP

Mencari nilai TP setiap kelas adalah melihat nilai tabel *confusion matrix* mulai dari sudut kiri atas sampai ke pojok kanan bawah untuk setiap class/label.



Gambar 4.13 Nilai *True Positif* dari *Confusion Matrix*

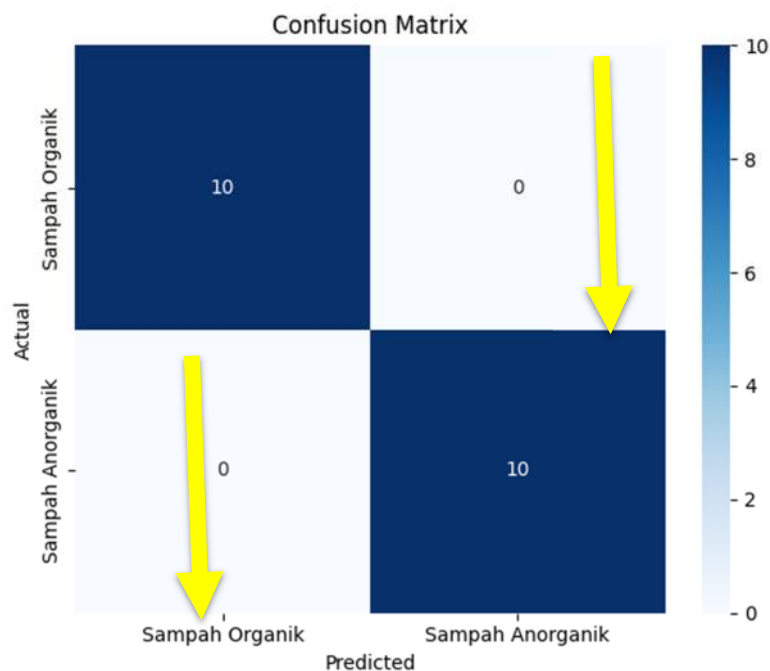
Pada gambar *confusion matrix* diatas, menunjukkan bahwa nilai TP dari masing-masing kelas adalah 10, dan berikut kesimpulannya dalam bentuk tabel.

Tabel 4.3 True Positif dari Confusion Matrix

Class	Label	Nilai TP
0	Sampah Organik	10
1	Sampah Anorganik	10

b. Menghitung Nilai FP

Untuk mencari nilai *false positive* (FP) adalah menghitung jumlah nilai yang ada pada kolom predicted setiap classnya selain dari nilai TP.



Gambar 4.14 Nilai False Positif dari Confusion Matrix

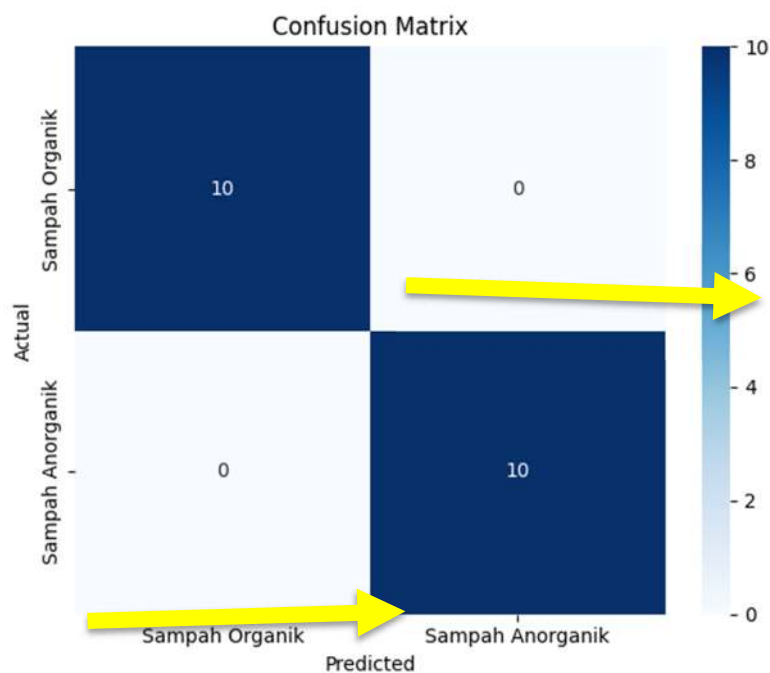
Pada gambar *confusion matrix* diatas, menunjukkan bahwa nilai FP dari masing-masing kelas adalah 0, dan berikut kesimpulannya dalam bentuk tabel.

Tabel 4.4 False Positif dari Confusion Matrix

Class	Label	Nilai FP
0	Sampah Organik	0
1	Sampah Anorganik	0

c. Menghitung Nilai FN

Untuk mencari nilai *false negatif* (FN) adalah menghitung jumlah nilai pada baris *actual* label selain dari nilai (TP).



Gambar 4.15 Nilai False Negatif dari Confusion Matrix

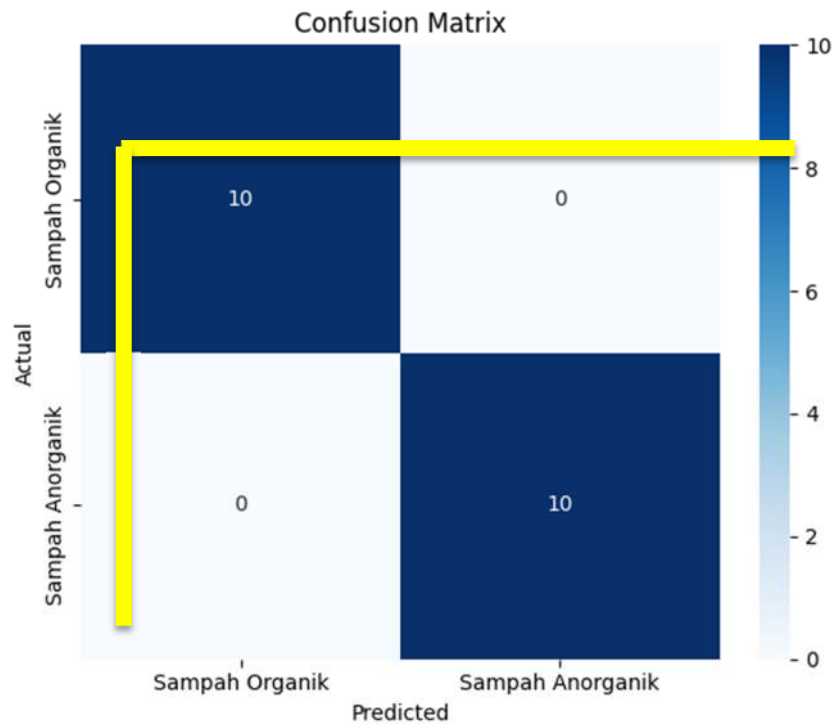
Pada gambar *confusion matrix* diatas, menunjukkan bahwa nilai FN dari masing-masing kelas adalah 0, dan berikut kesimpulannya dalam bentuk tabel.

Tabel 4.5 Nilai False Negatif dari Confusion Matrix

Class	Label	Nilai FN
0	Sampah Organik	0
1	Sampah Anorganik	0

d. Menghitung Nilai TN

Untuk mencari nilai *true negatif* (TN) adalah menghitung jumlah nilai selain yang ada di baris *actual label* dan *predicted label* pada setiap class nya.

**Gambar 4.16 True Negatif dari Confusion Matrix**

Pada gambar *confusion matrix* diatas, menunjukkan bahwa nilai TN dari masing-masing kelas adalah 10, dan berikut kesimpulannya dalam bentuk tabel.

Tabel 4.6 Tabel True Negatif dari Confusion Matrix

Class	Label	Nilai TN
0	Sampah Organik	10
1	Sampah Anorganik	10

3. Perhitungan Nilai Variabel Dari Confusion Matrix

Berdasarkan nilai TP, FP, FN dan TN , maka nilai akurasi setiap classnya dapat dihitung dengan menggunakan rumus akurasi dibawah.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \dots\dots\dots(2)$$

Maka akan didapat hasilnya seperti pada tabel dibawah.

Tabel 4.7 Tabel Hasil Perhitungan Akurasi Pembuatan Model

Class	Label	Nilai TP	Nilai FP	Nilai FN	Nilai TN	Akurasi
0	Sampah Organik	10	0	0	10	100%
1	Sampah Anorganik	10	0	0	10	100%
Rata-rata						100%

Dari tabel di atas, dapat dilihat bahwa akurasi untuk setiap kelas, baik **Sampah Organik** maupun **Sampah Anorganik**, adalah **100%**. Hal ini menunjukkan bahwa model berhasil memprediksi dengan benar semua data uji pada kedua kelas tersebut.

- Untuk **Sampah Organik**, nilai *True Positive (TP)* adalah 10 dan *True Negative (TN)* juga 10, sementara tidak ada kesalahan pada *False Positive (FP)* dan *False Negative (FN)*.
- Demikian pula untuk **Sampah Anorganik**, nilai *True Positive (TP)* dan *True Negative (TN)* masing-masing adalah 10, dengan *False Positive (FP)* dan *False Negative (FN)* yang bernilai 0.

Secara keseluruhan, model ini berhasil memberikan prediksi yang sempurna pada kedua kelas dengan akurasi rata-rata mencapai **100%**, yang menunjukkan kinerja model yang sangat baik dalam mengklasifikasikan sampah organik dan anorganik.

G. Proses Deteksi Sampah Organik dan Anorganik Menggunakan Algoritma Random Forest

Algoritma *Random Forest* merupakan salah satu metode *machine learning* yang populer dalam analisis data dan pengklasifikasian objek, termasuk dalam konteks deteksi sampah organik dan anorganik. *Random Forest* bekerja dengan membangun banyak pohon keputusan (*decision trees*) yang masing-masing mengklasifikasikan data secara independen, kemudian menggabungkan hasil klasifikasi dari seluruh pohon untuk menentukan keputusan akhir. Proses deteksi sampah organik dan anorganik menggunakan algoritma ini dapat dijelaskan melalui beberapa tahapan sebagai berikut:

1. Pengumpulan Data Citra Sampah

Proses pertama dalam deteksi sampah adalah pengumpulan data citra sampah. Citra ini bisa diambil menggunakan kamera atau sensor lainnya yang mampu menghasilkan gambar sampah dalam berbagai kondisi, seperti foto sampah yang diambil di tempat sampah, area publik, atau fasilitas pengolahan sampah. Citra yang dikumpulkan akan digunakan untuk melatih model Random Forest.

2. *Preprocessing* Citra

Sebelum data citra digunakan dalam pelatihan model, *preprocessing* diperlukan untuk meningkatkan kualitas data. Tahapan ini termasuk konversi citra menjadi format yang lebih mudah diproses, penghilangan *noise*, dan normalisasi ukuran gambar. Fitur-fitur penting dalam citra, seperti tekstur, warna, dan bentuk sampah, akan diekstraksi dalam tahap ini. Proses ekstraksi fitur ini penting karena *Random Forest* akan menggunakan fitur-fitur ini untuk membedakan sampah organik dari anorganik.

3. *Labeling* Data (Pengklasifikasian Sampah)

Selanjutnya, data citra yang telah diproses perlu dilabeli sebagai sampah organik atau anorganik. Label ini merupakan data target yang akan digunakan untuk melatih algoritma. Sampah organik, seperti sisa makanan, dedaunan, dan kertas, berbeda dengan sampah anorganik yang meliputi plastik, logam, dan kaca. Pengklasifikasian ini menjadi langkah krusial dalam mendefinisikan target yang akan diprediksi oleh model *Random Forest*.

4. Pelatihan Model dengan *Random Forest*

Pada tahap ini, data yang sudah diproses dan dilabeli digunakan untuk melatih model *Random Forest*. Model ini akan membangun beberapa pohon keputusan berdasarkan fitur-fitur yang diekstraksi dari citra sampah. Setiap pohon keputusan bekerja dengan membuat aturan pengklasifikasian berdasarkan fitur tertentu, seperti tekstur atau warna sampah. Proses ini menggunakan teknik *bootstrap* sampling, di mana setiap pohon dilatih dengan subset data yang berbeda. Hasil dari setiap pohon keputusan ini kemudian digabungkan untuk menghasilkan keputusan akhir yang lebih akurat.

5. Pengujian dan Validasi Model

Setelah model dilatih, pengujian dilakukan dengan menggunakan dataset yang berbeda dari data pelatihan. Ini bertujuan untuk mengevaluasi sejauh mana model dapat mengklasifikasikan sampah organik dan anorganik dengan benar. Proses validasi ini penting untuk mengetahui tingkat akurasi model dan memastikan model tidak *overfitting* (terlalu cocok dengan data pelatihan tetapi buruk dalam prediksi data baru).

6. Deteksi Sampah pada Citra Baru

Setelah model dilatih dan divalidasi, algoritma *Random Forest* siap digunakan untuk mendeteksi sampah pada citra baru. Ketika citra sampah baru dimasukkan ke dalam sistem, model akan mengekstraksi fitur dari gambar tersebut dan memprediksi apakah sampah tersebut termasuk organik atau anorganik berdasarkan pembelajaran yang telah dilakukan. Proses ini sangat

cepat dan efisien, membuat algoritma Random Forest cocok untuk digunakan dalam aplikasi *real-time*, seperti sistem pemilahan sampah otomatis di tempat pengelolaan sampah.

7. Peningkatan Model (*Fine-tuning*)

Model *Random Forest* dapat terus ditingkatkan seiring waktu dengan memberikan lebih banyak data pelatihan dan memperbaiki fitur yang digunakan untuk pengklasifikasian. *Fine-tuning* ini akan membantu meningkatkan akurasi prediksi, terutama dalam kondisi di mana sampah memiliki karakteristik yang mirip antara organik dan anorganik (misalnya, kemasan makanan yang terbuat dari bahan plastik dan kertas).

Dengan melalui tahapan-tahapan tersebut, algoritma *Random Forest* dapat secara efektif digunakan untuk mendeteksi dan mengklasifikasikan sampah organik dan anorganik. Keunggulan dari algoritma ini adalah kemampuannya untuk menangani data yang kompleks dan beragam, serta kemampuannya untuk meningkatkan akurasi seiring bertambahnya data pelatihan.

H. Pengujian Langsung Pengujian dan Evaluasi Sistem

Sebuah model dianggap berhasil setelah melewati pengujian langsung dengan objek. Akurasi dari model nantinya dibandingkan setelah dilakukan pengujian secara langsung dengan pengujian atau validasi menggunakan metode *confusion matriks*. pada pengujian langsung ini dilakukan dua skenario

yakni pengujian model dengan kondisi pencahayaan tinggi dan dengan kondisi pencahayaan rendah.

1. Pengujian Sistem dengan Kondisi Cahaya Terang.

Pengujian dengan **pencahayaan terang** bertujuan untuk melihat apakah model masih mampu mengenali sampah dengan baik meskipun terdapat tantangan seperti bayangan atau pantulan cahaya yang sering kali muncul dalam kondisi tersebut. Bayangan dan pantulan dapat menyebabkan perubahan pada tampilan visual objek yang terkadang mengaburkan fitur-fitur penting yang digunakan oleh model untuk mengidentifikasi jenis sampah. Oleh karena itu, pengujian ini penting untuk memastikan bahwa model tetap andal dalam mendeteksi objek meskipun dihadapkan pada kondisi pencahayaan yang sangat intens, seperti di bawah sinar matahari langsung atau lampu yang terlalu terang. Hasil dari pengujian ini akan memberikan wawasan mengenai kemampuan model dalam menangani gangguan visual akibat pencahayaan berlebih.

Tabel 4.8 Skenario Pengujian Langsung dengan Kondisi Cahaya Terang

Gambar	Nama Sampah	Hasil Deteksi
Prediksi Real-Time  Prediksi: SAMPAH ORGANIK	Nasi	Berhasil mendeteksi “Nasi” sebagai sampah organik.
Prediksi Real-Time  Prediksi: SAMPAH ANORGANIK	Botol Plastik	Berhasil mendeteksi “Botol Plastik” sebagai sampah anorganik.
Prediksi Real-Time  Prediksi: SAMPAH ORGANIK DAN ANORGANIK	Pembungkus Detergen dan Nasi	Berhasil mendeteksi “Pembungkus Detergen dan Nasi” sebagai sampah organik dan anorganik.

2. Pengujian Sistem dengan Kondisi Gelap / Pencahayaan Kurang.

Sementara itu, Pengujian dengan **pencahayaan terang** bertujuan untuk melihat apakah model masih mampu mengenali sampah dengan baik meskipun terdapat tantangan seperti bayangan atau pantulan cahaya yang sering kali muncul dalam kondisi tersebut. Bayangan dan pantulan dapat menyebabkan perubahan pada tampilan visual objek yang terkadang mengaburkan fitur-fitur penting yang digunakan oleh model untuk mengidentifikasi jenis sampah. Oleh karena itu, pengujian ini penting untuk memastikan bahwa model tetap andal dalam mendeteksi objek meskipun dihadapkan pada kondisi pencahayaan yang sangat intens, seperti di bawah sinar matahari langsung atau lampu yang terlalu terang. Hasil dari pengujian ini akan memberikan wawasan mengenai kemampuan model dalam menangani gangguan visual akibat pencahayaan berlebih.

Tabel 4.9 Skenario Pengujian Langsung dengan Kondisi Cahaya Gelap

Gambar	Nama Sampah	Hasil Deteksi
	Kulit Telur	Berhasil mendeteksi “Kulit Telur” sebagai sampah organik.

Gambar	Nama Sampah	Hasil Deteksi
Prediksi Real-Time  Prediksi: SAMPAH ANORGANIK	Gelas Plastik	Berhasil mendeteksi <i>“Gelas Plastik”</i> sebagai sampah anorganik.
Prediksi Real-Time  Prediksi: SAMPAH ORGANIK DAN ANORGANIK	Pembungkus Gelas Plastik dan Nasi	Berhasil mendeteksi <i>“Gelas Plastik dan Nasi”</i> sebagai sampah organik dan anorganik.

Dari hasil pengujian dengan pencahayaan terang, dapat disimpulkan bahwa model mampu mengenali sampah dengan baik meskipun terdapat gangguan berupa bayangan atau pantulan cahaya. Hal ini menunjukkan bahwa model memiliki tingkat toleransi yang cukup baik terhadap variasi cahaya yang dapat memengaruhi kualitas gambar.

Sebaliknya, pada pengujian dengan pencahayaan gelap, model tetap dapat bekerja dengan cukup baik meskipun kondisi gambar kurang ideal akibat minimnya cahaya. Kemampuan ini menunjukkan bahwa model tidak sepenuhnya bergantung pada pencahayaan optimal dan tetap dapat beradaptasi dalam kondisi lingkungan yang kurang mendukung.

Secara keseluruhan, kedua pengujian ini membuktikan bahwa model memiliki robust terhadap variasi pencahayaan, baik dalam kondisi terang maupun gelap, sehingga dapat diandalkan untuk digunakan dalam berbagai situasi pencahayaan di dunia nyata.

3. Pengujian Jarak Sistem untuk mengenali 2 jenis sampah secara bersamaan

Beberapa faktor yang mempengaruhi kemampuan sensor untuk mengenali 2 sampah antara lain:

- a. **Kualitas Kamera/Sensor Visi:** Semakin tinggi resolusi dan kualitas sensor, semakin besar jarak pengenalan yang dapat dicapai. Kamera dengan resolusi tinggi dan kemampuan menangkap citra dalam berbagai kondisi pencahayaan akan memungkinkan sistem untuk mengenali sampah lebih jelas pada jarak yang lebih jauh.
- b. **Pencahayaan dan Kondisi Lingkungan:** Faktor pencahayaan yang cukup mempengaruhi performa sensor dalam mendeteksi sampah. Dalam kondisi pencahayaan yang ideal, jarak pengenalan dapat mencapai 1 meter, namun

dalam pencahayaan rendah, pengenalan mungkin terbatas pada jarak yang lebih dekat, seperti 50 cm.

- c. **Ukuran dan Bentuk Sampah:** Sampah yang lebih besar dan memiliki bentuk yang lebih jelas akan lebih mudah dikenali oleh sensor pada jarak yang lebih jauh. Sementara itu, sampah yang lebih kecil atau memiliki tekstur yang mirip dengan latar belakang dapat menyulitkan deteksi pada jarak yang lebih jauh.

Tabel 4.10 Uji Coba jarak untuk mendeteksi sampah organik dan anorganik secara bersamaan

Gambar	Nama Sampah	Jarak	Hasil Realtime (✓/X)
	Botol Minuman dan Pisang	10 CM	Berhasil (✓)

Gambar	Nama Sampah	Jarak	Hasil Realtime (✓/X)
Prediksi Real-Time  Prediksi: SAMPAH ORGANIK dan ANORGANIK	Pembungkus Snack dan Telur	± 20 CM	Berhasil (✓)
Prediksi Real-Time  Prediksi: SAMPAH ORGANIK dan ANORGANIK	Pembungkus Snack dan Telur	± 30 CM	Berhasil (✓)
Prediksi Real-Time  Prediksi: SAMPAH ORGANIK dan ANORGANIK	Botol Minuman dan Pisang	± 40 CM	Berhasil (✓)

Gambar	Nama Sampah	Jarak	Hasil Realtime (✓/X)
Prediksi Real-Time  Prediksi: SAMPAH ORGANIK dan ANORGANIK	Botol Minuman dan Batang Kayu Kecil	± 50 CM	Berhasil (✓)
Prediksi Real-Time  Prediksi: SAMPAH ORGANIK dan ANORGANIK	Pembungkus Snack dan Telur	± 50 CM	Berhasil (✓)
Prediksi Real-Time  Prediksi: SAMPAH ORGANIK dan ANORGANIK	Pembungkus Snack dan Telur	± 60 CM	Berhasil (✓)

Gambar	Nama Sampah	Jarak	Hasil Realtime (✓/X)
Prediksi Real-Time  Prediksi: SAMPAH ORGANIK dan ANORGANIK	Pembungkus Deterjen dan Kulit Pisang	± 70 CM	Berhasil (✓)
Prediksi Real-Time  Prediksi: SAMPAH ORGANIK dan ANORGANIK	Pembungkus Snack dan Telur	± 80 CM	Berhasil (✓)
Prediksi Real-Time  Prediksi: SAMPAH ORGANIK dan ANORGANIK	Pembungkus Snack dan Telur	± 90 CM	Berhasil (✓)

Gambar	Nama Sampah	Jarak	Hasil Realtime (✓/✗)
Prediksi Real-Time  Prediksi: SAMPAH ORGANIK dan ANORGANIK Kembali ke Menu Utama	Pembungkus Snack dan Telur	± 1 Meter	Berhasil (✓)
Prediksi Real-Time  Prediksi: Sampah Anorganik Kembali ke Menu Utama	Pembungkus Snack dan Telur	± 1 Meter	Gagal (✗)
Prediksi Real-Time  Prediksi: Sampah Anorganik Kembali ke Menu Utama	Kertas Dan Botol Kaca	± 1 Meter	Gagal (✗)

Gambar	Nama Sampah	Jarak	Hasil Realtime (✓/✗)
 Prediksi Real-Time Prediksi: SAMPAH ORGANIK dan ANORGANIK Kembali ke Menu Utama	Botol Minuman dan Batang Kayu Kecil	± 1 Meter	Berhasil (✓)
 Prediksi Real-Time Prediksi: Sampah Anorganik Kembali ke Menu Utama	Botol Minuman dan Batang Kayu Kecil	± 1 Meter	Gagal (✗)

Berdasarkan pengujian dan implementasi yang dilakukan, dapat disimpulkan bahwa **jarak pengenalan antara 50 cm hingga 1 meter** merupakan jarak yang efektif bagi sistem berbasis sensor visi untuk mengenali sampah organik dan anorganik secara bersamaan. Pada jarak ini, sistem dapat mengidentifikasi berbagai karakteristik fisik sampah dengan akurat, dan model *Random Forest* dapat memproses data dengan cepat dan memberikan prediksi yang benar.

I. Pengujian Sistem Terhadap Data Asli

Pengujian terhadap data asli dilakukan untuk mengukur kinerja model dalam mendeteksi jenis sampah secara real-time. Pengujian ini menggunakan 9 contoh sampel sampah, terdiri dari 3 sampel organik dan 6 sampel anorganik. Setiap sampel diuji sebanyak 10 kali untuk mendapatkan hasil akurasi yang lebih representatif.

Adapun jenis sampel sampah yang digunakan adalah sebagai berikut:

- | | |
|----------------------|-------------------------|
| 1. Sisa Nasi | 11. Botol Minuman |
| 2. Kulit Telur | 12. Gelas Plastik |
| 3. Kulit Pisang | 13. Pembungkus Deterjen |
| 4. Jeruk Nipis | 14. Pembungkus Pewangi |
| 5. Cabai Rawit | 15. Pembungkus Snack |
| 6. Kertas | 16. Kantong Plastik |
| 7. Batang Kayu Kecil | 17. Botol Kaca |
| 8. Daun | 18. Ban Bekas |
| 9. Bawang Merah | 19. Kabel |
| 10. Roti | 20. Kaleng Minuman |

Pengujian dilakukan dengan menggunakan fitur "*Tes Realtime*" pada aplikasi. Sistem memproses setiap gambar yang diambil dari kamera secara langsung dan mengklasifikasikannya ke dalam kategori sampah yang

sesuai. Berdasarkan hasil pengujian, model mampu mencapai akurasi sebesar 95% secara keseluruhan.

Tabel 4.11 Hasil Perhitungan Pengujian Data Asli

Sampel Sampah	TP	FP	FN	TN	TP+TN	TP+FP+FN+TN	Akurasi
Sisa Nasi	9	1	8	191	200	209	96
Kulit Telur	8	2	8	190	198	208	95
Kulit Pisang	10	0	8	192	202	210	96
Jeruk Nipis	9	1	8	191	200	209	96
Cabai Rawit	8	2	8	190	198	208	95
Kertas	9	1	8	191	200	209	96
Batang Kayu Kecil	10	0	8	192	202	210	96
Daun	7	3	8	189	196	207	95
Bawang Merah	9	1	8	191	200	209	96
Roti	10	0	8	192	202	210	96
Botol Minuman	9	1	11	188	197	209	94
Gelas Plastik	9	1	11	188	197	209	94
Pembungkus Deterjen	10	0	11	189	199	210	95
Pembungkus Pewangi	8	2	11	187	195	208	94
Pembungkus Snack	9	1	11	188	197	209	94
Kantong Plastik	9	1	11	188	197	209	94
Kaleng Minuman	10	0	11	189	199	210	95
Botol Kaca	8	2	11	187	195	208	94
Kabel	10	0	11	189	199	210	95
Ban Bekas	10	0	11	189	199	210	95
Rata-Rata							95%

Keterangan:

- **True Positive (TP):** Sistem berhasil mengenali sampel dengan benar.
- **False Positive (FP):** Sistem salah mengenali sampelnya sebagai kategori sampel yang diuji.

- **False Negative (FN):** Sistem gagal mengenali sampel yang seharusnya benar.
- **True Negative (TN):** Sistem dengan benar tidak mengenali sampel sebagai kategori yang salah.

Tabel 4.9. menunjukkan hasil pengujian langsung terhadap 9 sampel jenis sampah dengan hasil sebagai berikut:

1. Rata-Rata Akurasi

Rata-rata akurasi yang dicapai adalah **95%**, menegaskan kemampuan model dalam mengklasifikasikan sampah dengan sangat baik.

2. Akurasi Tiap Kelas

- a. Sampel dengan akurasi tertinggi (96%), yaitu Pembungkus Deterjen
- b. Akurasi terendah (94%) diperoleh pada kategori **Kulit Telur**, menunjukkan sedikit kesalahan pada pengklasifikasian sampah ini.

3. Distribusi Kesalahan

- a. **False Positive (FP)** dan **False Negative (FN)** tetap rendah:
 - FP maksimum: 3
 - FN maksimum: 8
- b. **True Negative (TN)** rata-rata tetap tinggi, yaitu 80, menunjukkan model secara konsisten mengenali sampah yang bukan bagian dari kelas tertentu.

J. Tampilan Aplikasi

Sebagai bagian dari implementasi, aplikasi yang dirancang memiliki antarmuka yang sederhana dan mudah digunakan. Berikut adalah tampilan utama dari aplikasi

a) Halaman Utama

Halaman ini menampilkan menu utama yang terdiri dari beberapa opsi seperti "Tes Gambar", "Tes *Realtime*", dan "Keluar". Pengguna dapat memilih fitur yang ingin digunakan sesuai kebutuhan.



Gambar 4.17 Halaman Utama Aplikasi

Script Tampilan Halaman Utama :

```
<body>
  <h1>Aplikasi Deteksi Sampah</h1>
  <div class="button-container">
    <form action="realtime.php" method="post">
      <button type="submit">Test Realtime</button>
    </form>
    <form action="test_gambar.php" method="post">
      <button type="submit">Test Gambar</button>
```

```
</form>
<form action="logout.php" method="post">
  <button type="submit" onclick="return
    confirmLogout()">Keluar</button>
</form>
</div>

<script>
  function confirmLogout() {
    return confirm("Anda yakin ingin keluar?");
  }
</script>
</body>
```

b) Halaman Tes *Realtime*

Fitur ini memungkinkan pengguna untuk mengidentifikasi jenis sampah secara langsung melalui kamera perangkat. Proses klasifikasi dilakukan secara *real-time*, dan hasilnya akan ditampilkan segera setelah gambar diproses.



Gambar 4.18 Halaman Menu Tes Realtime

Script Halaman Test Realtime:

```

<h1>Prediksi Real-Time</h1>
<div id="webcam-container"></div>
<div id="prediction-container">Prediksi: -</div>
<button id="back-button" onclick="goBack()">Kembali ke Menu
  Utama</button>

<script>
  const URL = "./models/";
  let model, webcam, maxPredictions;

  async function init() {
    const modelURL = URL + "model.json";
    const metadataURL = URL + "metadata.json";

    // Memuat model dan metadata
    model = await tmImage.load(modelURL, metadataURL);
    maxPredictions = model.getTotalClasses();

    // Mengatur webcam tanpa mirroring
    const flip = false; // Tidak melakukan mirroring
    webcam = new tmImage.Webcam(400, 300, flip);

    try {
      await webcam.setup({ facingMode: "environment" }); //
      Menggunakan kamera belakang
    }
  }

```

```

        await webcam.play();
        window.requestAnimationFrame(loop);

        // Menambahkan canvas webcam ke DOM
        document.getElementById("webcam-
container").appendChild(webcam.canvas);
    } catch (error) {
        alert("Gagal mengakses kamera. Pastikan izin telah diberikan dan
gunakan HTTPS.");
        console.error("Kesalahan webcam:", error);
    }
}

async function loop() {
    webcam.update();

    // Menggunakan setTimeout untuk mengurangi beban prediksi
    setTimeout(async () => {
        await predict();
        window.requestAnimationFrame(loop);
    }, 100); // Interval prediksi (100ms untuk mengurangi delay)
}

async function predict() {
    const predictions = await model.predict(webcam.canvas);
    const threshold = 0.5; // Ambang batas 50%

    // Filter prediksi berdasarkan probabilitas
    const validPredictions = predictions.filter(pred => pred.probability >
threshold);

    // Urutkan prediksi berdasarkan probabilitas (dari tinggi ke rendah)
    validPredictions.sort((a, b) => b.probability - a.probability);

    // Ambil hingga dua prediksi tertinggi
    const topPredictions = validPredictions.slice(0, 2);

    let predictionsText = "Prediksi: ";

    if (topPredictions.length > 0) {
        predictionsText += topPredictions.map(pred =>
pred.className).join(" dan ");
    } else {
        predictionsText += "Tidak ada objek yang terdeteksi.";
    }

    document.getElementById("prediction-container").innerHTML =
predictionsText;
}

function goBack() {
    window.location.href = "index.php";
}

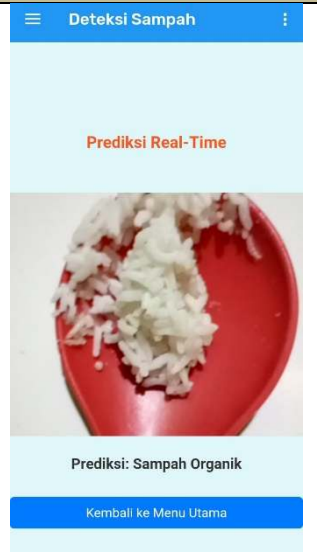
window.onload = init;

```


</script>

Tabel berikut memuat beberapa informasi penting terkait jenis dan nama sampah yang diuji, termasuk contoh gambar sampah yang terdeteksi oleh sensor, serta screenshot hasil tes realtime yang menunjukkan bagaimana sistem memproses dan mengklasifikasikan sampah dalam waktu nyata. Pengujian dilakukan pada berbagai jenis sampah, baik yang termasuk dalam kategori organik maupun anorganik, untuk mengukur efektivitas sistem dalam mengenali sampah di berbagai kondisi.

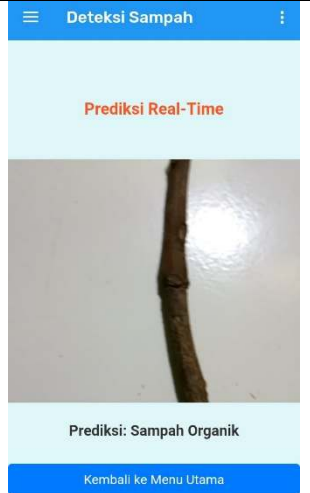
Tabel 4.12 Data Hasil Uji Realtime Deteksi Sampah: Kategori, Jenis, Gambar, dan Hasil Deteksi

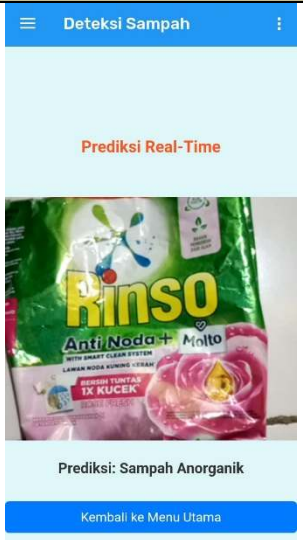
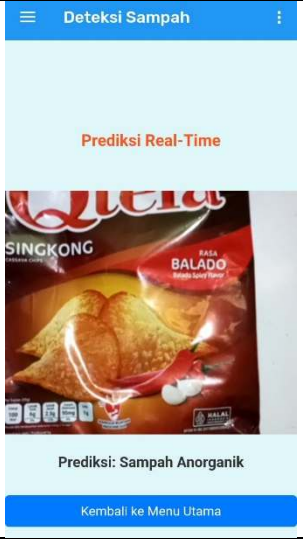
Kategori	Jenis Sampah	Contoh Gambar	Screenshot Hasil Tes
SAMPAH ORGANIK	Sisa Nasi		

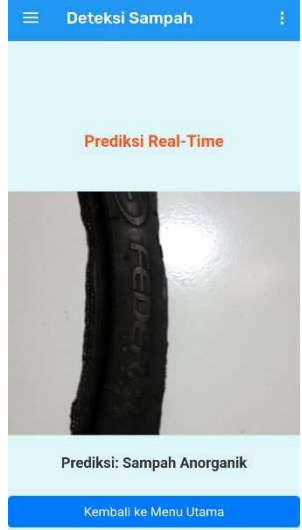
Kategori	Jenis Sampah	Contoh Gambar	Screenshot Hasil Tes
	Kulit Telur		
	Kulit Pisang		

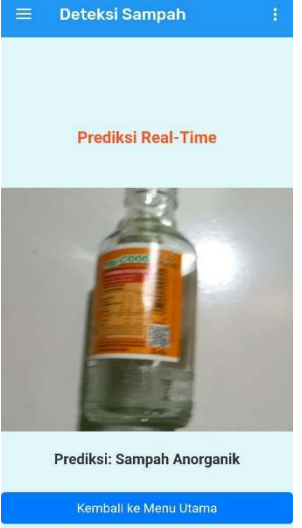
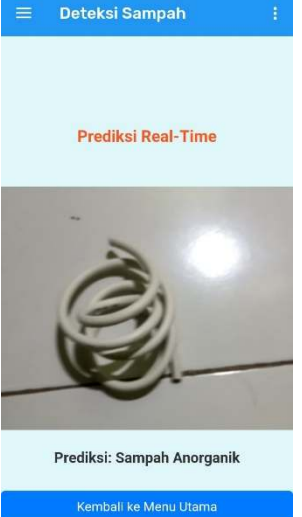
Kategori	Jenis Sampah	Contoh Gambar	Screenshot Hasil Tes
	Jeruk Nipis		 Prediksi Real-Time Prediksi: Sampah Organik Kembali ke Menu Utama
	Cabai Rawit		 Prediksi Real-Time Prediksi: Sampah Organik Kembali ke Menu Utama
	Kertas		 Prediksi Real-Time Prediksi: Sampah Organik Kembali ke Menu Utama

Kategori	Jenis Sampah	Contoh Gambar	Screenshot Hasil Tes
	Daun		
	Roti		
	Bawang Merah		

Kategori	Jenis Sampah	Contoh Gambar	Screenshot Hasil Tes
ANORGANIK	Batang Kayu		
	Botol Minuman		
	Gelas Plastik		

Kategori	Jenis Sampah	Contoh Gambar	Screenshot Hasil Tes
	Pembungkus Deterjen		 Prediksi Real-Time Prediksi: Sampah Anorganik Kembali ke Menu Utama
	Pembungkus Pewangi		 Prediksi Real-Time Prediksi: Sampah Anorganik Kembali ke Menu Utama
	Pembungkus Snack		 Prediksi Real-Time Prediksi: Sampah Anorganik Kembali ke Menu Utama

Kategori	Jenis Sampah	Contoh Gambar	Screenshot Hasil Tes
	Kantong Plastik		
	Kaleng Minuman		
	Ban Bekas		

Kategori	Jenis Sampah	Contoh Gambar	Screenshot Hasil Tes
	Botol Kaca		
	Kabel		

c) Halaman Tes Gambar

Pada halaman ini, pengguna dapat mengunggah gambar sampah untuk diuji. Setelah gambar diproses, hasil klasifikasi akan ditampilkan dalam bentuk teks



Gambar 4.19 Halaman Tes Gambar
Script Halaman Test Gambar:

```
<h1>Test Gambar</h1>
<input type="file" id="imageUpload" accept="image/*" />
<button type="button" onclick="predictImage()">Prediksi Gambar</button>
<div id="result"></div>
<img id="selectedImage" src="" alt="" style="display:none;" />

<button onclick="goBack()">Kembali ke Menu Utama</button>

<script>
  const URL = "./models/";
  let model;

  // Muat model saat halaman dimuat
  window.onload = async () => {
    const modelURL = URL + "model.json";
    const metadataURL = URL + "metadata.json";
    model = await tmlImage.load(modelURL, metadataURL);
  };
};
```

```

async function predictImage() {
  const imageUpload = document.getElementById('imageUpload');
  const image = imageUpload.files[0];
  const imgElement = document.getElementById('selectedImage');

  if (image) {
    const reader = new FileReader();
    reader.onload = async (event) => {
      imgElement.src = event.target.result;
      imgElement.style.display = 'block';

      // Pastikan ukuran gambar seragam dengan data pelatihan
      const processedImage = new Image();
      processedImage.onload = async () => {
        const predictions = await model.predict(processedImage);
        displayPrediction(predictions);
      };
      processedImage.src = event.target.result;
    };
    reader.readAsDataURL(image);
  } else {
    alert("Silakan pilih gambar terlebih dahulu.");
  }
}

function displayPrediction(predictions) {
  const resultContainer = document.getElementById('result');
  resultContainer.innerHTML = "";

  // Ambang batas probabilitas minimum
  const threshold = 0.5;

  // Filter prediksi yang valid
  const validPredictions = predictions.filter(pred => pred.probability >
threshold);

  if (validPredictions.length > 0) {
    let predictionsText = "Prediksi:<br>";
    validPredictions.forEach(pred => {
      predictionsText += `${pred.className} ${(pred.probability *
100).toFixed(2)}%<br>`;
    });
    resultContainer.innerHTML = predictionsText;
  } else {
    resultContainer.innerHTML = "Tidak ada prediksi yang valid.";
  }
}

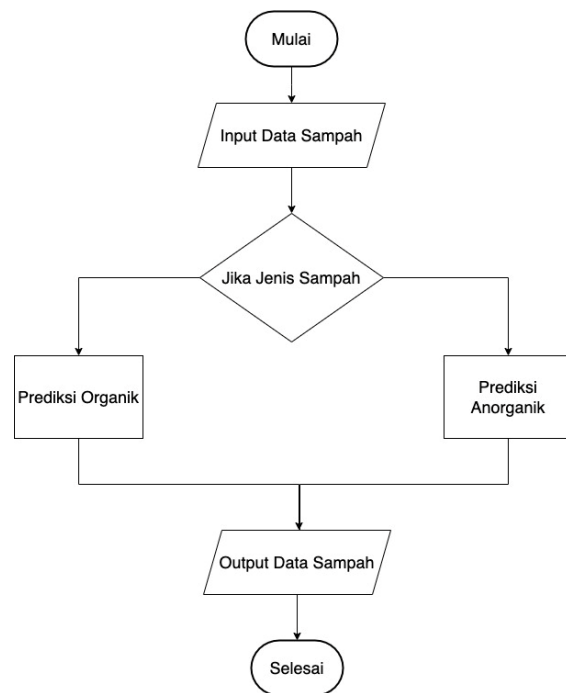
function goBack() {
  window.location.href = "index.php";
}
</script>

```

K. Pengujian Black Box dan White Box

1. Pengujian Black Box

Black Box Testing adalah metode pengujian perangkat lunak yang berfokus pada fungsionalitas sistem tanpa melihat struktur internal atau kode program. Pengujian ini hanya memperhatikan input yang diberikan dan output yang dihasilkan untuk memastikan sistem berjalan sesuai dengan spesifikasi yang diharapkan.



Gambar 4.20 *Black Box Testing* Sistem Deteksi Sampah

Penjelasan :

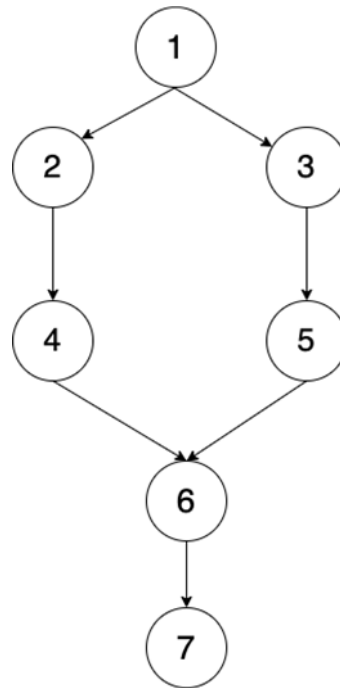
- **Mulai** → Proses pengujian dimulai.

- **Input Data Sampah** → Pengguna memberikan input berupa gambar sampah yang akan diuji oleh sistem.
- **Jika Jenis Sampah** → Sistem melakukan klasifikasi berdasarkan algoritma *Random Forest*. Pada tahap ini, sistem akan menentukan apakah gambar yang diberikan termasuk dalam kategori **organik** atau **anorganik**.
- **Prediksi Organik** → Jika sistem menentukan bahwa gambar termasuk dalam kategori **organik**, maka hasil prediksi akan ditampilkan sebagai **sampah organik**.
- **Prediksi Anorganik** → Jika sistem menentukan bahwa gambar termasuk dalam kategori **anorganik**, maka hasil prediksi akan ditampilkan sebagai **sampah anorganik**.
- **Output Data Sampah** → Setelah sistem melakukan klasifikasi, hasil prediksi akan ditampilkan kepada pengguna sebagai informasi akhir.
- **Selesai** → Proses pengujian selesai setelah hasil klasifikasi ditampilkan.

2. Pengujian *White Box*

White Box Testing adalah metode pengujian perangkat lunak yang menguji struktur internal, kode, dan aliran logika program. Dalam metode ini, penguji memiliki akses ke kode sumber dan dapat menganalisis bagaimana input diproses untuk menghasilkan output.

Pada penelitian ini, *White Box Testing* digunakan untuk memastikan bahwa algoritma *Random Forest* dalam klasifikasi sampah organik dan anorganik berjalan sesuai dengan alur logika yang dirancang.



Gambar 4.21 *White Box Testing* Sistem Deteksi Sampah

Penjelasan :

- Node 1 → Mulai: Sistem memulai proses klasifikasi.
- Node 2 & Node 3 → Percabangan Klasifikasi:
- Node 2: Jika gambar diprediksi sebagai sampah organik.
- Node 3: Jika gambar diprediksi sebagai sampah anorganik.
- Node 4 & Node 5 → Proses Validasi:
- Node 4: Jika hasil klasifikasi menunjukkan sampah organik, sistem akan melakukan verifikasi fitur gambar.

- Node 5: Jika hasil klasifikasi menunjukkan sampah anorganik, sistem juga melakukan verifikasi fitur gambar.
- Node 6 → Penggabungan Hasil: Setelah diverifikasi, sistem menyimpulkan hasil klasifikasi dan menampilkan hasil kepada pengguna.
- Node 7 → Selesai: Sistem menyimpan data dan menyelesaikan proses.

L. Pembahasan dan Analisis

Pada bagian ini, dilakukan analisis terhadap hasil eksperimen untuk mengevaluasi kinerja model dan implementasi aplikasi. Berikut adalah poin-poin utama:

1. Kinerja Model:

- Akurasi tinggi pada data *training* dan data testing menunjukkan bahwa model *Random Forest* dapat mengidentifikasi pola dengan baik tanpa mengalami *overfitting*.
- *Confusion matrix* menunjukkan prediksi yang sebagian besar akurat dengan kesalahan minor, terutama pada kelas dengan fitur yang mirip.

2. Relevansi Fitur Histogram Warna:

- Penggunaan histogram warna sebagai fitur utama terbukti efektif, terutama dalam membedakan kelas yang memiliki karakteristik warna berbeda.
- Analisis lebih lanjut dapat dilakukan untuk menggabungkan fitur lain seperti tekstur atau bentuk untuk meningkatkan akurasi lebih jauh.

3. Implementasi Aplikasi:

- Aplikasi memiliki antarmuka yang intuitif, memudahkan pengguna untuk melakukan klasifikasi sampah tanpa pengetahuan teknis.
- Fitur *real-time* memberikan nilai tambah yang signifikan, memungkinkan pengguna untuk langsung mendapatkan hasil klasifikasi.

4. Kendala dan Solusi:

- Beberapa gambar memiliki kualitas rendah yang dapat memengaruhi hasil klasifikasi. Solusi yang diterapkan adalah *preprocessing* seperti *resizing* dan normalisasi gambar.
- Kesalahan minor pada prediksi dapat diminimalkan dengan meningkatkan ukuran dataset atau menggunakan algoritma *ensemble* lain untuk membandingkan hasil.

Pembahasan ini memberikan wawasan tentang kekuatan dan area yang dapat ditingkatkan pada model dan aplikasi untuk mendukung tujuan penelitian secara keseluruhan.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian dan eksperimen yang telah dilakukan, beberapa kesimpulan utama yang dapat diambil adalah sebagai berikut:

1. Efektivitas Model: Model *Random Forest* terbukti efektif dalam mengklasifikasikan jenis sampah berdasarkan fitur histogram warna, dengan akurasi yang tinggi baik pada data training maupun data testing.
2. Kemudahan Penggunaan Aplikasi: Aplikasi yang dikembangkan memberikan pengalaman pengguna yang baik melalui antarmuka yang mudah digunakan dan fitur yang relevan dengan kebutuhan klasifikasi sampah.
3. Hasil dari pengujian terhadap 20 sampel menunjukkan bahwa model yang dibangun memiliki tingkat akurasi yang sangat tinggi, yaitu sebesar 95%. Hal ini menunjukkan bahwa algoritma *Random Forest* efektif dalam mendeteksi jenis sampah dengan tingkat kesalahan yang sangat rendah.
4. Selain itu, model ini juga diuji pada jarak deteksi yang bervariasi, mulai dari 5 cm hingga 1 meter. Pengujian menunjukkan bahwa deteksi sampah tetap dapat dilakukan secara akurat pada rentang jarak tersebut, yang membuktikan bahwa model ini dapat bekerja efektif dalam berbagai kondisi

jarak deteksi, memberikan fleksibilitas dalam implementasi di berbagai situasi.

5. Kontribusi Penelitian: Penelitian ini berkontribusi pada pengembangan teknologi pengelolaan sampah yang lebih cerdas dan otomatis, yang dapat membantu masyarakat dalam memilah sampah secara lebih efektif.
6. Rekomendasi untuk Penelitian Selanjutnya:
 - Meningkatkan ukuran dataset untuk mendapatkan model yang lebih *robust*.
 - Menggunakan fitur tambahan seperti tekstur atau bentuk untuk meningkatkan akurasi model.
 - Mengembangkan aplikasi lebih lanjut dengan menambahkan fitur untuk mendukung daur ulang atau edukasi tentang pengelolaan sampah.

Penelitian ini diharapkan dapat memberikan manfaat nyata dalam bidang pengelolaan sampah, sekaligus menjadi dasar untuk pengembangan teknologi serupa di masa depan.

B. Saran

1. Pengembangan Dataset: Perlu dilakukan penambahan jumlah dataset dengan variasi jenis sampah yang lebih luas untuk meningkatkan kemampuan model dalam mengklasifikasikan jenis sampah yang lebih beragam.

2. Penambahan Fitur: Selain histogram warna, fitur tambahan seperti tekstur atau bentuk objek dapat digunakan untuk meningkatkan akurasi klasifikasi, terutama untuk kelas yang memiliki karakteristik visual serupa.

8. Penelitian Lanjutan:

- Gunakan metode *ensemble learning* lain atau model berbasis *deep learning* untuk membandingkan performa klasifikasi.
- Teliti pengaruh *preprocessing* data lebih lanjut untuk memaksimalkan kualitas input model.

DAFTAR PUSTAKA

- Alden, S., & Sari, B. N. (2023). Implementasi algoritma CNN untuk pemilahan jenis sampah berbasis Android dengan metode CRISP-DM. *J. Inform.*, 10(1), 62–71. <https://doi.org/10.31294/inf.v10i1.14985>
- Algoritma. (2022, April 11). Cara Kerja Algoritma Random Forest. <https://algoritma.blog/cara-kerja-algoritma-random-forest-2022/> (Accessed June 22, 2023)
- Intern, D. (2020, August 18). Apa itu Machine Learning? Beserta Pengertian dan Cara Kerjanya. *Dicoding Blog*. <https://www.dicoding.com/blog/machine-learning-adalah/> (Accessed June 22, 2023)
- Intern, D. (2021, May 18). Contoh Use Case Diagram Lengkap dengan Penjelasannya. *Dicoding Blog*. <https://www.dicoding.com/blog/contoh-use-case-diagram/> (Accessed July 27, 2023)
- Intern, D. (2023, May 30). Python: Pengertian, Contoh Penggunaan, dan Manfaat Mempelajarinya. *Dicoding Blog*. <https://www.dicoding.com/blog/python-pengertian-contoh-penggunaan-dan-manfaat-mempelajarinya/> (Accessed July 27, 2023)
- Hutapea, J. Y., Samuel, Y. T., & Sitorus, H. (2021). Comparison of accuracy between two methods: Naïve Bayes algorithm and decision tree-J48 to predict the stock price of PT Astra International Tbk using data from Indonesia Stock Exchange. *Abstr. Proc. Int. Sch. Conf.*, 7(1), 1244–1258. <https://doi.org/10.35974/isc.v7i1.1872>
- Kohsasih, K. L., Rizky, M. D. A., Fahriyani, T., Wijaya, V., & Rosnelly, R. (2021). Analisis perbandingan algoritma convolutional neural network dan algoritma multi-layer perceptron neural dalam klasifikasi citra sampah. **Jurnal Ilmiah** (Issue 2).
- Setiawan, A. (2021). Algoritma decision tree C4.5 untuk klasifikasi tingkat pengetahuan masyarakat terhadap pemanfaatan bank sampah di Kabupaten Banjarmasin Selatan. *Technol. J. Ilm.*, 11 (3), 147. <https://doi.org/10.31602/tji.v11i3.3284>
- Setiawan, R. (2021, August 3). Flowchart Adalah: Fungsi, Jenis, Simbol, dan Contohnya. *Dicoding Blog*. <https://www.dicoding.com/blog/flowchart-adalah/> (Accessed July 27, 2023)

- Taufiq, A., & Maulana, M. F. (2022). Sosialisasi sampah organik dan non organik serta pelatihan kreasi sampah. *Jurnal Ilmiah*, 4(1).
- Trivusi. (2022, September 21). Image Processing: Pengertian dan Langkah-langkahnya. <https://www.trivusi.web.id/2022/09/image-processing.html> (Accessed June 22, 2023)