

BAB III

METODOLOGI PENELITIAN

3.1 Gambaran Umum Objek Penelitian

Dalam penelitian ini, langkah awal yang dilakukan adalah dengan melakukan identifikasi terhadap latar belakang dan permasalahan yang sedang terjadi. Pada tahap ini, ditemukan bahwa sampah di lingkungan sekitar merupakan hal yang perlu diperhatikan secara lebih karena rendahnya kesadaran masyarakat yang salah tempat sampah ketika hendak membuang sampah. Oleh karena itu sebagai solusi untuk permasalahan tersebut, pada penelitian ini akan membuat klasifikasi gambar sampah berdasarkan bahannya untuk meningkatkan pengetahuan masyarakat untuk melakukan pemilahan sampah dan melakukan daur ulang sampah [142].

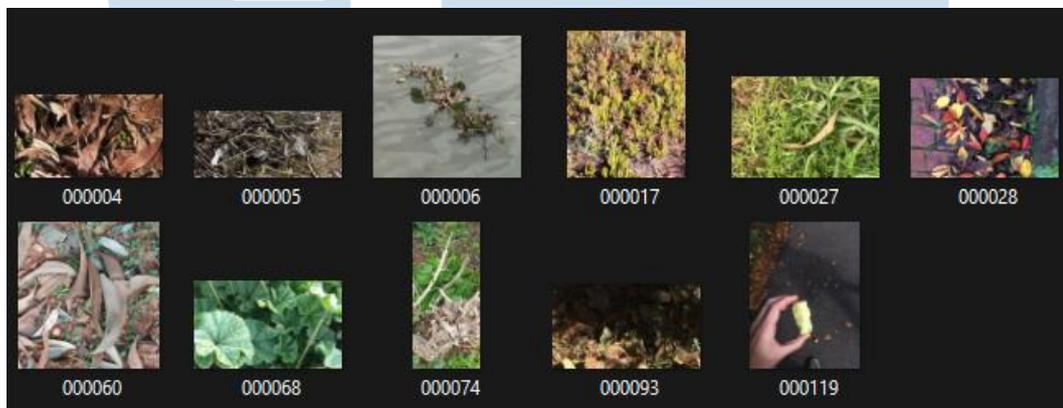


Gambar 3.1 Ilustrasi klasifikasi sampah organik dan non-organik

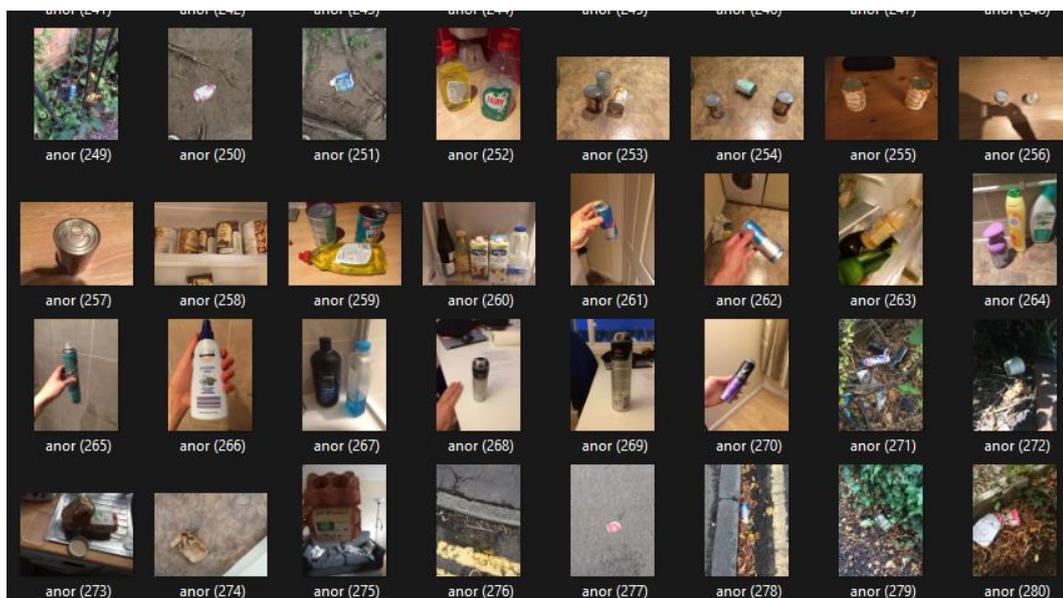
Sumber: [143]

Gambaran umum objek penelitian berfokus pada klasifikasi gambar sampah berdasarkan jenisnya, yaitu sampah organik dan non-organik, seperti ilustrasi yang terlihat ilustrasi gambar 3.1. Untuk mendukung efektivitas pemilahan dan pengolahan sampah, dapat dibantu dengan bantuan mesin dan penglihatan komputer (*computer vision*) dalam melakukan deteksi objek pada gambar, untuk meningkatkan pengetahuan masyarakat dalam melakukan klasifikasi sampah sesuai

dengan jenisnya, yaitu sampah organik dan sampah non-organik. Sumber dataset yang digunakan untuk tahapan *training* dalam penelitian ini berasal dari dataset TACO dan dataset garbage data yang berasal dari web *Kaggle*. Dataset TACO berisi 1.500 gambar sampah yang ditemukan di lingkungan, dan diambil menggunakan kamera *handphone* pada tahun 2020, dengan *free copyright licenses* untuk penggunaan umum [144]. Pada gambar 3.2, merupakan tampilan *dataset* gambar organik, dan gambar 3.3 merupakan tampilan gambar sampah non-organik dari data sampah TACO.

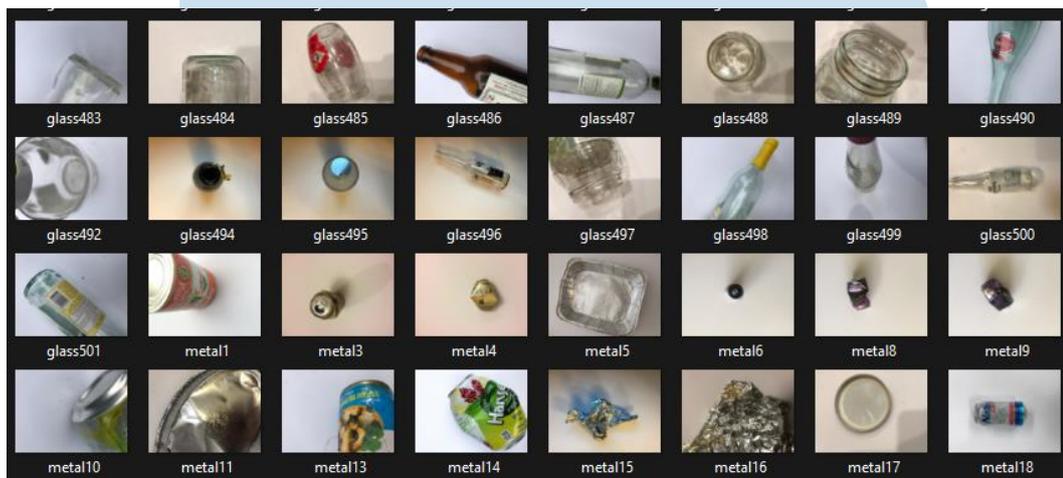


Gambar 3.2 Dataset TACO sampah organik



Gambar 3.3 Dataset TACO sampah non-organik

Pada dataset *garbage_data*, merupakan campuran dataset yang telah dibagi menjadi data *training* dan data *validation*, dengan total 20.125 gambar sampah. Dataset *garbage* terdiri dari beberapa jenis sampah, yaitu sampah karton, kertas, besi, plastic, gelas, organik, dan sampah secara umum (*trash*). Pada gambar 3.4, merupakan gambar data sampah non-organik dari dataset *garbage* data, sedangkan untuk gambar 3.5 merupakan gambar sampah organik dari dataset *garbage* data.



Gambar 3.4 Dataset *garbage* data sampah non-organik



Gambar 3.5 Dataset *garbage* data sampah organik

Sedangkan untuk *testing*, digunakan sebanyak 100 gambar sampah, yang terdiri dari 50 gambar sampah organik dan non-organik yang diambil langsung dengan menggunakan kamera *handphone* peneliti seperti contoh pada gambar 3.6, dan 50 sisanya merupakan gambar sampah organik dan non-organik yang diambil

langsung dengan menggunakan kamera *laptop* peneliti, seperti pada gambar 3.7. Kedua jenis kamera digunakan sebagai perbandingan keakuratan prediksi model terhadap gambar dan resolusi yang berbeda.



Gambar 3.6 *Sample* gambar lapangan dengan kamera *handphone*

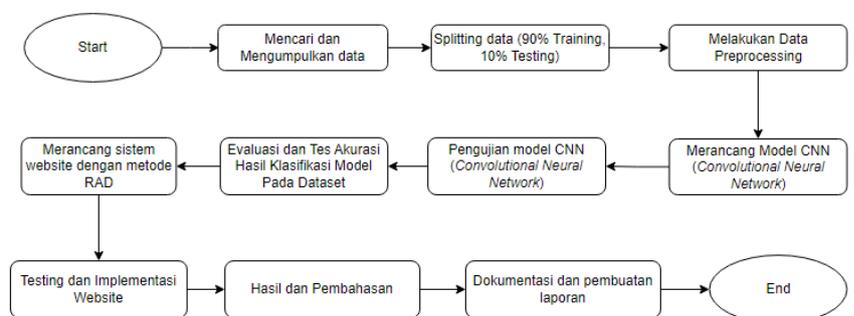


Gambar 3.7 *Sample* gambar lapangan dengan kamera *laptop*

3.2 Metode Penelitian

Jenis penelitian yang digunakan dalam penelitian ini adalah jenis penelitian kuantitatif. Penelitian kuantitatif merupakan metode yang didasarkan pada perhitungan analisis statistik untuk mengetahui hubungan antar variabel [145].

3.2.1 Alur Penelitian



Gambar 3.8 Alur Penelitian

Dalam mengupayakan tujuan diadakan penelitian, terdapat beberapa tahapan yang perlu dilalui, seperti alur penelitian pada gambar 3.8 yang menggambarkan proses analisis data dengan metode *deep learning*, yaitu algoritma *Convolutional Neural Network* (CNN), karena performanya yang baik dalam mengklasifikasikan gambar. *Software* yang digunakan adalah *software Anaconda Navigator* dengan ekstensi *Jupyter Notebook*, *package Tensorflow* dan Keras. Adapun dasar referensi pembuatan alur penelitian ini merujuk pada penelitian sebelumnya oleh Islahatul Inayanti (2022) yang menggunakan alur serupa dalam penelitian tentang sampah organik dan sampah non-organik, namun dengan menggunakan *tools google collab* untuk melatih model [146].

Algoritma CNN terdiri dari beberapa lapisan karena merupakan algoritma dalam pembelajaran *deep learning*, CNN mengklasifikasikan gambar dengan serangkaian operasi dengan beberapa lapisan konvolusi hingga akhirnya membentuk suatu pola. Lapisan dalam arsitektur CNN terdiri dari lapisan *convolution layer*, *max pooling layer*, *flatten*, *dense layer*, dan *output*. Seluruh lapisan dibagi menjadi *feature learning* dan *classification* (tahap klasifikasi). Adapun berikut merupakan penjelasan tahapan yang ditempuh:

1. Mencari dan mengumpulkan data

Data yang dikumpulkan adalah data yang sesuai dengan lingkup permasalahan sampah yang terjadi di lingkungan sekitar sebagai bahan penelitian. Data yang digunakan adalah data gambar sampah TACO dan dataset *garbage* yang dipublikasikan di situs web *kaggle*. *Website Kaggle* merupakan situs untuk berbagi ide dan penelitian, didukung dengan beragam dataset yang dapat diakses secara gratis.

Dataset TACO diunduh dengan jumlah 1,500 gambar sampah non-organik dan organik sebelum dibersihkan. Dataset TACO dipilih karena data tersebut merupakan data gambar sampah dengan kualitas baik yang diambil dari tangkapan layar kamera *handphone*. Namun, setelah melalui tahap pembersihan secara manual, ternyata banyak gambar yang duplikat dan kurang relevan. Pada akhirnya, data gambar dari dataset TACO yang

digunakan hanya sebanyak 627 gambar dengan kategori sampah non-organik dan hanya ditemukan sebanyak 13 gambar sampah organik.

Dikarenakan jumlah yang sangat sedikit dan kurang efektif untuk *training data*, dataset kedua yaitu dataset *garbage* dipilih sebagai pelengkap data sampah organik yang dibutuhkan untuk penelitian. Dataset *garbage* diunduh dengan jumlah awal 20.125 data gambar sampah organik dan non-organik. Setelah dataset dibersihkan, hanya sebanyak 14.960 data gambar sampah yang digunakan, dikarenakan ditemui beberapa gambar yang tidak relevan dan kurang berfokus pada objek. Seperti latar belakang yang terlalu luas, dan cenderung mendistraksi fokus objek pada gambar.

Kedua dataset tersebut kemudian digabung untuk melatih model CNN, sehingga didapatkan total sebesar $14.960 + 640 = 15.600$ data sampah. Kemudian, untuk gambar yang diambil secara langsung dari lapangan berjumlah 100 gambar yang terdiri dari 50 gambar sampah organik dan 50 sampah non-organik, dengan sumber kamera pengambilan gambar yang berbeda, yaitu kamera *smartphone*, dan kamera *laptop*.

2. *Splitting Data*

Training (80%)	Validation (10%)	Testing (10%)
----------------	------------------	---------------

Gambar 3.9 Rancangan *splitting data*

Splitting dataset yang berjumlah 15.600 data gambar dibagi menjadi dua bagian, yaitu data non-organik dan data sampah organik. Adapun untuk pembagiannya adalah dengan persentase 80:10:10 dalam pembagian *training*, *validation*, dan *testing* seperti yang terlihat pada gambar 3.9. Sebanyak 12.480 data digunakan untuk *training*, 1.560 data *validation*, dan 1.560 sisanya adalah data *test* untuk menguji model. Pembagian data ini dilakukan berdasarkan referensi penelitian serupa dengan penggunaan algoritma CNN mendeteksi penyakit daun oleh Yazid et al. (2023), yang memperoleh akurasi mencapai angka sempurna pada angka 100% dengan

menerapkan skema pembagian data tersebut pada model *DenseNet169* sebagai bagian dari arsitektur CNN [147].

3. **Data Preprocessing**

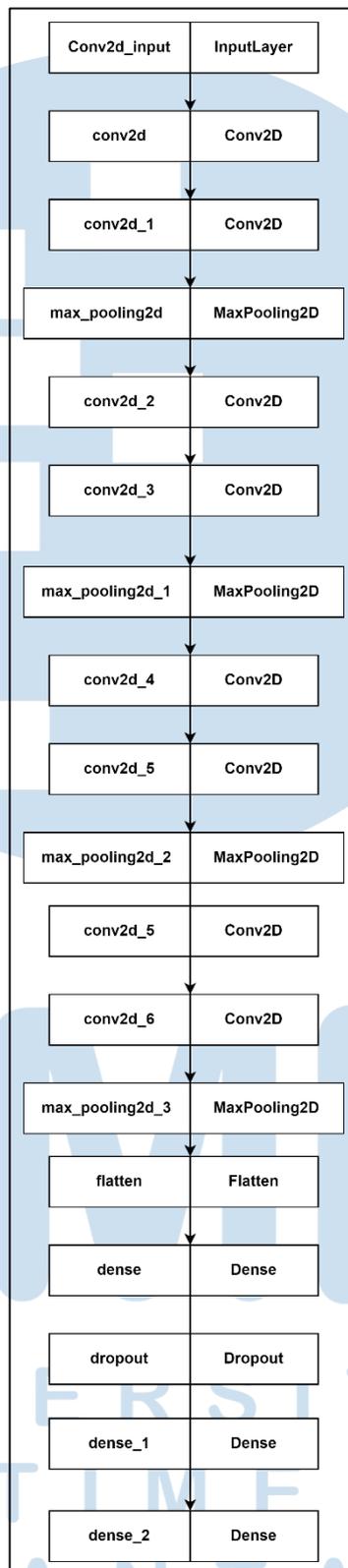
Langkah selanjutnya yang ditempuh adalah data *preprocessing*, yaitu tahap penyiapan data sebelum data siap untuk di proses. Dalam *preprocessing* ini, dilakukan proses *resize* yang bertujuan untuk menyetarakan ukuran dan *format* data (penjelasan lebih lanjut mengenai *resize* terlampir pada sub-bab 2.1.4). Data gambar pada dataset kemudian akan diolah dengan *Image Generator* dari *library Keras*.

4. **Perancangan Model CNN (*Convolutional Neural Network*)**

Setelah data melalui tahap *preprocessing*, data siap digunakan untuk melatih model algoritma CNN. Pada tahapan ini, dilakukan perancangan model untuk pelatihan data sehingga model dapat belajar dan mengenali objek pada gambar. Model terdiri dari 11 *hidden layer*, yaitu 7 *layer* sebagai tahap konvolusi dan 4 *layer* sebagai tahap *max pooling*. Kemudian disusul dengan 1 *layer flatten*, 1 *layer dropout*, dan 3 *layer dense*.

Dalam *layer flatten*, data *array* yang berbentuk multi-dimensi diubah menjadi sebuah vektor agar dapat dimasukkan ke dalam *layer fully connected*. Sebelum dimasukkan, dilakukan fungsi *dropout* yang bermanfaat untuk mencegah terjadinya *overfitting* (melakukan pengurangan terhadap neuron secara acak). Rancangan model CNN dapat dilihat pada gambar 3.10.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.10 Tahap perancangan model CNN

Setelah model CNN selesai dirancang, data *training* yang telah dikumpulkan tadi kemudian digunakan untuk melatih model. Akurasi model diukur untuk mendapatkan model terbaik dari proses klasifikasi. Model terbaik kemudian akan disimpan untuk diujikan pada data *test* pada tahapan selanjutnya. Pada proses pengujian model, dibuat sebuah iterasi (*epoch*), untuk menentukan berapa kali model akan melakukan proses *training*. Pada proses pelatihan ini juga digunakan fungsi *loss* untuk mengukur performa model dalam mengklasifikasikan target gambar. Fungsi *loss* untuk mentranslasikan model ketika model gagal memberikan hasil yang diinginkan, karena adanya kesalahan yang perlu diperhatikan [148].

5. Pengujian Model CNN

Sesudah model CNN selesai dibuat, model akan diuji pada data *test*, serta data lapangan untuk menguji ketepatan dan kebenaran model dalam mengklasifikasikan gambar sampah yang di input, apakah termasuk ke dalam kelas organik ataupun kelas non-organik. Apabila hasil akurasi cenderung tinggi, maka model merupakan model yang baik dan terbukti dapat melakukan klasifikasi sampah dengan akurat.

6. Evaluasi dan Tes Akurasi Model CNN

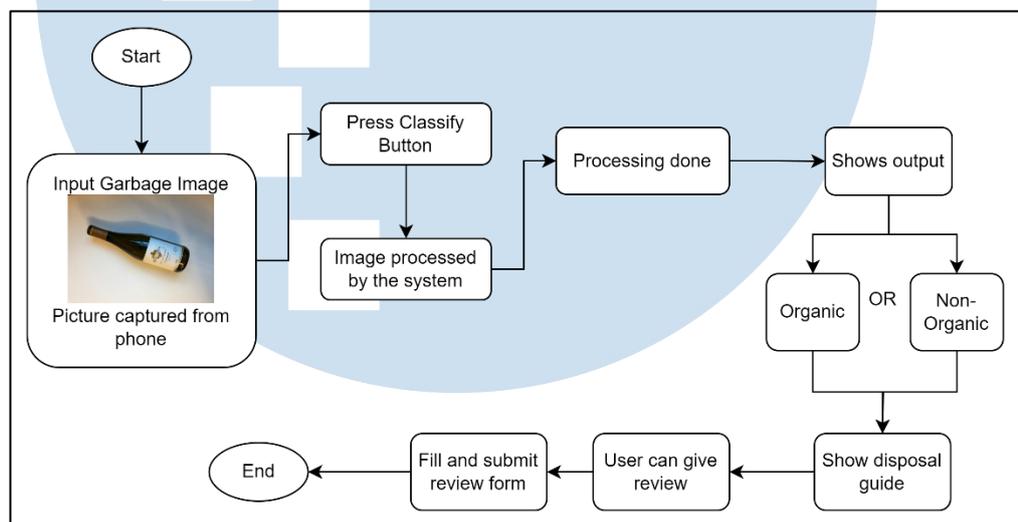
Evaluasi model dilakukan dengan menggunakan *confusion matrix* untuk mengukur tingkat akurasi hasil klasifikasi gambar. Dari nilai *confusion matrix* yang diperoleh, dapat dihitung performa dari matriks yang digunakan, yaitu nilai akurasi, *f1-score*, *precision*, dan *recall*.

7. Merancang Sistem Website dengan Metode RAD

Langkah berikutnya adalah merancang sebuah *website* sebagai luaran untuk mengimplementasikan algoritma CNN. *Website* dibangun dengan menggunakan metode pengembangan sistem RAD (*Rapid Application Development*). Perancangan *website* dibantu dengan *framework bootstrap* melalui *software Visual Studio Code*, dengan bahasa pemrograman *html*. Adapun alasan pemilihan *website* sebagai luaran dari

penelitian ini adalah karena banyaknya keunggulan (dijelaskan lebih lanjut pada bab sebelumnya yaitu bab 2.1.6). Salah satunya seperti kemudahan akses lintas platform (komputer, *laptop*, *tablet*, maupun *smartphone*), sehingga dapat diakses kapan dan dimana saja. Berbeda dengan *android* yang hanya terbatas pada platform *smartphone*.

Website yang sudah selesai dirancang akan dihubungkan model CNN yang terbaik sebagai model yang akan melakukan klasifikasi sampah organik maupun sampah non-organik berdasarkan gambar yang di *input* oleh pengguna *website* nantinya.



Gambar 3.11 Rancangan alur kerja *website*

Rancangan *website* yang dimaksud seperti alur kerja pada gambar 3.11, yang dimulai dengan *user* memasukkan gambar sampah sebagai *input*, dapat berasal dari gambar yang diambil langsung dari *user* ataupun diambil dari *gallery user*, kemudian menekan tombol *classify*, lalu gambar akan diproses oleh sistem. Setelah selesai, sistem akan menunjukkan sebuah *output* berupa sampah organik ataupun sampah non-organik. Kemudian, menunjukkan informasi yang relevan sesuai dengan kategori klasifikasi sampah yang muncul. Jika sampah termasuk sampah organik, maka informasi pembuangan sampah ke tempat sampah organik akan muncul, dan sebaliknya. *User* kemudian dapat memberikan *review* berdasarkan hasil klasifikasi yang didapat, kemudian mengisi form *review* dan melakukan

submit sehingga form tersebut akan tersimpan dalam sistem dan terlihat oleh *admin* atau *developer* sebagai bentuk *feedback* atas hasil klasifikasi yang dilakukan oleh sistem.

8. Testing dan Implementasi sistem klasifikasi gambar pada *website*

Dilakukan testing *website* yang telah selesai dibuat, lalu melakukan pengujian untuk memastikan bahwa semua fungsi berjalan sesuai skenario jalannya sistem. Hal ini bertujuan untuk memeriksa apakah masih ada kesalahan ataupun *error* pada sistem klasifikasi gambar.

9. Hasil dan pembahasan

Tahapan terakhir, hasil rancangan model CNN dan *website* akan dibuatkan pembahasan. Selain itu, juga didokumentasikan kesimpulan yang diperoleh dari penelitian yang telah dilakukan dan saran dari peneliti agar penelitian dapat diperbaiki maupun dikembangkan pada penelitian selanjutnya.

10. Dokumentasi dan Pembuatan Laporan

Hasil dari perancangan sistem akan didokumentasikan ke dalam laporan untuk menggambarkan kegiatan yang telah ditempuh selama penelitian, serta hasil yang didapatkan dari penelitian.

3.2.2 Metode Penyelesaian

Untuk mengurangi volume tumpukan sampah di Indonesia, peneliti membuat sebuah sistem klasifikasi gambar sampah, dengan menggunakan algoritma CNN. Adapun alasan pemilihan algoritma CNN adalah karena selain banyak diimplementasikan dalam *task* klasifikasi gambar pada banyak penelitian *machine learning*, juga merupakan algoritma klasifikasi gambar yang memang dibuat secara khusus untuk melakukan pemrosesan gambar dalam ranah *deep learning* sekalipun [149]. Adapun pada tabel 3.1 merupakan perbandingan algoritma CNN dengan dua algoritma klasifikasi gambar yang umum digunakan dalam ranah *machine learning*, yaitu SVM [150] dan KNN [151], serta algoritma klasifikasi gambar dalam ranah *deep learning*, yaitu ANN [152] dan RNN [153].

Tabel 3.1 Perbandingan Algoritma CNN dengan Algoritma Lain

Algoritma	Kegunaan	Kelebihan	Kekurangan
CNN (<i>Convolutional Neural Network</i>)	Melakukan klasifikasi gambar dengan cara ekstraksi fitur	<p>Memiliki kemampuan yang baik dalam mengolah gambar.</p> <p>Menggunakan lebih dari satu dimensi sehingga tidak mudah kehilangan informasi <i>spatial</i> yang akan diekstraksi sebagai informasi.</p> <p>Dapat mengatasi variasi data dalam segi pencahayaan, rotasi, skala, dan pergeseran objek dalam gambar.</p>	<p>Membutuhkan banyak data pelatihan dalam jumlah besar.</p> <p>Memakan waktu dan membutuhkan sumber daya komputasi yang kuat dalam melakukan pelatihan model pada data yang besar.</p> <p>Arsitektur yang kompleks dan tidak mudah dipahami</p>
SVM (<i>Support Vector Machine</i>)	Melakukan klasifikasi gambar dan pengenalan objek pada gambar dengan mencari <i>hyperplane</i> terbaik dalam ruang n-dimensi dalam mengklasifikasikan titik data	<p>Efektif untuk data berdimensi tinggi karena memiliki ruang fitur dengan dimensi tinggi.</p> <p>Memiliki generalisasi yang baik pada data baru sehingga dapat mengurangi resiko <i>overfitting</i>.</p> <p>Mampu menangani data linear dan nonlinear dengan memperluas ke ruang fitur yang lebih tinggi.</p>	<p>Sensitif terhadap <i>noise</i>, seperti <i>outlier</i> atau data yang tidak tepat sehingga model yang dihasilkan kurang baik.</p> <p>Performa kurang baik pada data yang besar.</p> <p>Sensitif terhadap pemilihan parameter, jika pemilihan tidak tepat dapat menyebabkan <i>overfitting</i> atau <i>underfitting</i>.</p>

Algoritma	Kegunaan	Kelebihan	Kekurangan
KNN (<i>K-Nearest Neighbors</i>)	Melakukan klasifikasi gambar dengan mengambil jumlah nilai k terdekat (<i>neighbors</i>) sebagai acuan dalam menentukan kelas dari data baru.	Sederhana, adaptif terhadap perubahan data Pelatihan model tidak memakan waktu. Kinerja yang baik bahkan pada data yang rumit.	Sensitif terhadap data yang tidak relevan. Memerlukan pengaturan parameter- k secara tepat. Rentan terhadap data yang tidak seimbang.
ANN (<i>Artificial Neural Network</i>)	Melakukan klasifikasi gambar dengan cara mengambil fitur penting secara otomatis dan mempelajari pola yang kompleks dalam gambar.	Dapat menangani data dengan dimensi tinggi seperti gambar beresolusi tinggi. Kemampuan belajar adaptif dan dapat menangani data yang kompleks Toleransi tinggi terhadap data <i>training</i> yang tidak sempurna ataupun memiliki kesalahan	Membutuhkan jumlah data yang besar Sensitif dalam pemilihan parameter Memerlukan waktu dan sumber daya komputasi yang besar Model ANN rentan terhadap <i>overfitting</i>
RNN (<i>Recurrent Neural Network</i>)	Melakukan klasifikasi gambar dengan memproses urutan <i>pixel</i> atau fitur vector dari gambar sebagai <i>input</i> dan menghasilkan <i>output</i> berupa klasifikasi akhir urutan.	Memiliki sel memori yang dapat menyimpan informasi urutan gambar sebelumnya untuk membuat keputusan klasifikasi pada konteks sebelumnya. Menangkap setiap informasi dalam data <i>input</i> secara sekuensial Kuat dalam memproses data sekuensial seperti teks, audio, video	Memiliki permasalahan mempelajari hubungan jangka panjang dalam urutan data Permasalahan hilangnya gradien (<i>gradient vanishing or exploding</i>) dalam <i>training</i> jangka panjang Model RNN cenderung kompleks

Algoritma	Kegunaan	Kelebihan	Kekurangan
			dan tidak dapat diimplementasikan pada fitur yang tidak bergantung pada urutan

Dari ketiga algoritma pada tabel 3.1, setiap algoritma memiliki kekurangan dan kelebihan masing-masing. Namun, algoritma CNN dipilih untuk melakukan klasifikasi gambar dalam penelitian karena meskipun training membutuhkan banyak waktu, namun tingkat akurasi dan performanya lebih baik jika dibandingkan dengan algoritma SVM maupun KNN dalam mengelola data gambar. Sama halnya dengan ANN dan RNN yang memiliki model yang kompleks dan meskipun dapat melakukan klasifikasi gambar, implementasi kedua algoritma tersebut masih jarang digunakan dan kurang populer dalam ranah deteksi objek ataupun klasifikasi gambar karena bentuk model yang kompleks.

Algoritma CNN sebelumnya juga banyak digunakan pada penelitian sebelumnya untuk melakukan klasifikasi gambar, salah satunya penelitian berjudul “*Waste Image Classification Based on Transfer Learning and Convolutional Neural Network*” dan “*Application of Convolutional Neural Network based on transfer learning for garbage classification*” dengan tingkat akurasi diatas 80%.

3.2.3 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode RAD. Selain RAD, terdapat metode pengembangan sistem jangka pendeklainnya yang cukup umum digunakan, seperti *prototyping*. Namun, metode RAD dipilih karena menyesuaikan dengan kebutuhan dan kondisi yang ada. Adapun tabel 3.2 merupakan perbandingan dari metode RAD dengan *prototyping* dan SCRUM [127] [154].

Tabel 3.2 Tabel Perbandingan Metode Pengembangan Sistem

Faktor Pemanding	RAD	Prototyping	SCRUM
Skala proyek	Efektif pada proyek dengan skala kecil	Skala proyek kecil hingga besar	Efektif pada proyek dengan skala kecil
Perancangan model	Menggunakan komponen yang sudah ada	Membuat dari awal rancangan sistem	Melakukan perencanaan pada aktivitas yang akan dilakukan dalam merancang model.
Analisa Kebutuhan Sistem	Kebutuhan yang telah dibuat dapat dikurangi maupun diubah untuk menyesuaikan kebutuhan pengguna saat <i>testing</i> dijalankan.	Kebutuhan sistem dapat diubah maupun dikurangi pada saat proses <i>testing</i> .	Kebutuhan sistem dapat ditambah atau dikurangi pada saat dilakukan <i>sprint</i>
<i>Testing</i>	Dijalankan ketika rancangan telah terbentuk sehingga sistem dapat dievaluasi untuk kemudian dimasukkan sebagai bahan revisi.	Dijalankan saat sistem masih dalam bentuk <i>prototype</i> sehingga hasil <i>testing</i> bisa saja mengubah rancangan sistem.	Dijalankan secara iteratif di setiap <i>sprint</i> , sehingga dapat diubah menyesuaikan dengan fungsionalitas dan penerimaan pengguna.
Pendekatan	Iteratif dan sedikit demi sedikit. Siklus pendek dan berulang-ulang, fokus pengembangan komponen dapat diimplementasikan secara cepat	Iteratif, perbaikan desain didarkan pada masukan dari pengguna	Iteratif, perlu dilakukan <i>sprint</i> untuk memonitor perkembangan.
Proses pengembangan sistem	Cepat dan intensif, melibatkan pemahaman kebutuhan dan penyelesaian yang cepat	Cepat dan singkat dalam bentuk <i>prototype</i>	Cepat, lebih terstruktur dan cakupan yang lebih luas

Faktor Perbandingan	RAD	Prototyping	SCRUM
Waktu dan Biaya	Waktu singkat, namun estimasi biaya lebih sulit karena sifatnya iteratif.	Singkat dan hemat waktu	Singkat dan adaptif

Berdasarkan perbandingan dalam tabel 3.2, dilakukan pemilihan metode RAD untuk digunakan dalam penelitian ini karena menyesuaikan dengan kebutuhan durasi waktu dan kemudahan perancangan sistem sebagai objek penelitian. Dalam penelitian ini, metode RAD digunakan karena sistem yang dibuat cukup sederhana dan berskala kecil, sehingga tidak memerlukan waktu yang lama dan biaya tidak terlalu besar. Metode RAD yang digunakan dalam penelitian ini mencakup tiga tahapan utama, yaitu:

1. **Requirement**

Pada tahapan ini, dilakukan identifikasi kebutuhan sistem klasifikasi gambar sampah yaitu dengan mengumpulkan informasi berupa permasalahan yang terjadi di sekitar lingkungan dan kebutuhan yang dibutuhkan. Observasi dilakukan dengan mengamati objek penelitian yaitu sampah jenis organik dan non-organik.

2. **Workshop RAD Design**

Desain sistem dilakukan dengan menggunakan perencanaan dari pemodelan sistem seperti *use case diagram*, *activity diagram*, dan *class diagram*. Tujuan dari dibuatnya pemodelan tersebut yaitu agar *user* dapat memahami rancangan dan alur sistem yang dibuat oleh *developer*. Database juga dibuat sebagai tempat penyimpanan gambar yang telah di input oleh *user*, dan menyimpan hasil *review* dari *user*. Penyimpanan *database* menggunakan XAMPP dengan *database phpmyadmin* untuk menyimpan perancangan objek, implementasi model CNN, serta *file* yang digunakan untuk menampilkan *website*, seperti file HTML, *Python* dan CSS.

3. *Testing and Implementation*

Implementation dilakukan sebagai finalisasi perancangan sistem, kemudian dilakukan pengujian (*testing*) dengan metode *black-box testing*, dan *UAT (User Acceptance Testing)* untuk menguji fungsionalitas sistem apakah seluruh fungsi berjalan dengan benar atau tidak.

3.3 Teknik Pengumpulan Data

Data yang digunakan adalah data sekunder yang diambil dari *website Kaggle*, sebuah situs kumpulan data *open source*. Tautan dataset yang digunakan, yaitu dataset TACO (<https://www.kaggle.com/datasets/kneroma/tacotrashdataset>), yang terdiri dari 1500 data gambar sampah yang ditemukan di lingkungan sekitar, yang diambil menggunakan fitur kamera pada *handphone*.

Dataset kedua yang digunakan dalam penelitian adalah *garbage dataset* yang juga bersumber dari *web Kaggle*, dengan tautan sebagai berikut (<https://www.kaggle.com/datasets/besserwisser/garbage-dataset>). Dataset *garbage* berisi 20.125 gambar sampah organik dan non-organik yang telah dibagi dua, yaitu data *train* dan data *validation*. Kedua dataset tersebut kemudian dibersihkan, di seleksi dan digabungkan menjadi satu sebelum digunakan untuk proses pelatihan model.

Adapun penggabungan kedua dataset ini dilakukan karena dataset TACO lebih banyak mengandung konten sampah non-organik dibandingkan sampah organik, sehingga dibutuhkan dataset kedua sebagai pelengkap dataset organik yang masih kurang, agar dataset cukup untuk digunakan sebagai bahan latih untuk pengujian model. Hasil penggabungan kedua dataset berjumlah 15.600 gambar sampah.

3.3.1 Populasi dan Sample

Penelitian ini digunakan sejumlah populasi dan sampel, yaitu total keseluruhan gambar sampah yang telah dikategorikan berdasarkan jenisnya. Jenis yang dimaksud adalah sampah organik dan sampah non-organik, gambar sampah

yang dikumpulkan kemudian diurutkan dengan total 12.480 gambar untuk data *training* dan 1.560 sisanya untuk data *test* dari kedua jenis sampah. Adapun jumlah total data yang digunakan yaitu 15.600 data gambar sampah. Seluruh data dari populasi dan sampel akan diambil untuk kemudian dijadikan sebagai acuan dalam penelitian, tidak ada teknik *sampling*, semua populasi yang telah ditentukan sebelumnya akan dilibatkan dalam penelitian.

3.3.2 Periode Pengambilan Data

Dataset yang digunakan adalah dataset TACO dan dataset *garbage* yang bersumber dari *web Kaggle*. Periode pengambilan data gambar dataset TACO (*Trash Annotations in Context*) yaitu hanya pada satu periode, yaitu masa publikasi pertamanya pada tahun 2020, dengan total 1,500 gambar sampah. Sementara dataset kedua, yaitu dataset *garbage* dipublikasi pada Juli 2022, yang terdiri dari 20.125 gambar sampah organik dan non-organik, namun data diseleksi lagi hingga data yang digunakan hanya sebesar 15.600 total gambar.

3.4 Variabel Penelitian

Terdapat dua jenis variabel pada penelitian, yaitu variabel independen dan variabel dependen. Variabel Independen merupakan variabel yang memiliki pengaruh terhadap variabel lain. Di sisi lain, variabel dependen adalah variabel yang dipengaruhi oleh variabel independen [155].

3.4.1 Variabel Independen

Pada penelitian ini yang merupakan variabel independen yaitu gambar objek sampah.

3.4.2 Variabel Dependen

Pada penelitian ini yang merupakan variabel dependen adalah klasifikasi jenis sampah, sampah organik dan non-organik.

3.5 Teknik Analisis Data dan *Tools* Penelitian

Data yang perlu dianalisis adalah data gambar sampah. Kumpulan *dataset* data gambar sampah digunakan untuk melatih model algoritma dalam

mendeteksi objek dan melakukan klasifikasi sesuai dengan jenisnya, yaitu sampah organik maupun non-organik. Proses analisis data yang digunakan yaitu menggunakan salah satu metode dalam *deep learning* untuk melakukan klasifikasi gambar, yaitu algoritma *Convolutional Neural Network* (CNN).

Algoritma CNN akan melakukan klasifikasi gambar sampah dengan cara membentuk sebuah pola menggunakan proses operasi konvolusi yang terdiri atas beberapa layer utama, seperti layer *convolutional*, layer *pooling*, dan layer *dense*. Setelah model CNN selesai dibuat, model akan diimplementasikan ke sebuah situs *website* yang membolehkan penggunaanya mengunggah foto dari *handphone*, ataupun memilih gambar dari *gallery* terkait objek sampah yang ingin dijadikan sebagai data *input* untuk model dalam melakukan klasifikasi sampah jenis organik maupun non-organik.

Pengguna kemudian dapat memberikan komentar dan *review* terkait hasil klasifikasi model tersebut. Adapun beberapa pilihan *tools* yang dapat dimanfaatkan untuk mendukung data *preprocessing*, dan proses pelatihan model algoritma dalam melakukan analisis data, yaitu *Python*, *MATLAB* dan *R-Studio*. Pada tabel 3.3 adalah perbandingan dari ketiga *tools* tersebut [156].

Tabel 3.3 Perbandingan Python, MATLAB dan Rstudio

No.	Indikator Pembeding	Python	MATLAB	R-Studio
1	Kompleksitas <i>syntax</i>	Tidak terlalu sulit	Mudah dipelajari	Agak sulit
2	Skala pengolahan data	Besar	Besar	Kecil
3	Harga	Gratis	Lisensi berbayar	Gratis
4	Jumlah fitur	Lebih banyak	Lebih sedikit	Lebih sedikit
5	Platform	Windows, macOS, Linux	Windows, macOS, Linux	Windows, macOS, Linux
6	Performa	Cepat	Lebih lambat dari R	Lebih lambat dari <i>Python</i>
7	Ketersediaan	<i>Open-source</i>	Tidak <i>open-source</i>	<i>Open-source</i>

8	Integrasi dengan Bahasa pemrograman lain	Mudah dihubungkan, fleksibel	Mudah dihubungkan	Dapat dihubungkan, tapi tidak selengkap MATLAB ataupun <i>Python</i>
9	Fungsional	<i>Image processing, machine learning, deep learning, design website, programming</i>	<i>Machine learning, image processing, signal processing</i>	Hanya analisa statistik
10	Kemampuan analisis	Dapat melakukan menganalisa <i>multivariable</i>	Dapat melakukan menganalisa <i>multivariable</i>	Tidak dapat melakukan analisa <i>multivariable</i>

Berdasarkan tabel perbandingan *tools Python, MATLAB dan RStudio* pada tabel 3.3, *tools* yang dipilih dalam penelitian ini adalah *Python*. Selain terkenal sebagai yang menyediakan banyak *library* dan *framework* yang dapat diakses secara gratis, *python* juga memiliki komunitas yang besar, agar saat pengguna mengalami *error* atau kesulitan saat menggunakan *python*, dapat mencari solusi ataupun bertanya ke orang yang pernah mengalami masalah serupa.

Performa *python* yang lebih cepat dan bahasa pemrograman sederhana juga mendukung dalam melakukan penelitian *image recognition* dalam mendeteksi objek dan melakukan klasifikasi pada gambar. *Python* juga mendukung penggunaan *package Keras* dan *library TensorFlow*.

Selain *tools* yang digunakan untuk analisis, *tools* yang digunakan untuk membangun sistem klasifikasi gambar sampah berbasis *website* adalah dengan *text editor*, untuk menyambungkan *model* dari *python* untuk diimplementasikan ke *website* dengan bahasa pemrograman html. Berikut merupakan perbandingan dua aplikasi *text editor* yang umum digunakan dalam melakukan edit terhadap kode

teks, yaitu *Visual Studio Code*, dan *Sublime Text* [157] [158]. Tabel 3.4 merupakan perbandingan *text editor* [159].

Tabel 3.4 Perbandingan Visual Studio Code, dan Sublime Text

No.	Indikator Pembeding	<i>Visual Studio Code</i>	<i>Sublime Text</i>
1	Akses	Gratis	Gratis, namun lisensi berbayar
2	Pemakaian memory RAM	Lebih banyak	Lebih sedikit
3	Jumlah fitur	Banyak	Lebih sedikit
4	Tampilan	Menarik, sederhana dan modern	Sederhana
5	Performa	Lebih cepat	Agak lambat
6	Integrasi dengan <i>Git</i>	Terintegrasi langsung	Membutuhkan <i>plugin</i>
7	Penyelesaian <i>code</i>	Mendukung penyelesaian otomatis	Memerlukan instalasi <i>package</i>
8	Ekstensi	Tersedia, dapat mengelola <i>extension</i> dalam aplikasi	Tidak ada fitur untuk mengelola <i>extension</i> dalam aplikasi
9	EMMET (<i>auto-complete</i>)	Tersedia	Memerlukan ekstensi <i>plugin</i>
10	Komunitas	Besar dan efektif dengan banyak sumber daya dan <i>plugin</i>	Sedikit komunitas dan pilihan <i>plugin</i>

Berdasarkan tabel perbandingan *tools Visual Studio Code, Sublime Text* dan *Eclipse IDE* pada tabel 3.4, *tools* yang dipilih sebagai *editor code* adalah *Visual Studio code*, dengan fungsi bawaan yang mendukung dengan kelengkapan fiturnya. Sementara itu, *sublime text* memerlukan beberapa *plugin* dan ekstensi yang harus di-*install* terlebih dahulu sebelum dapat menggunakan fitur yang serupa dengan *Visual Studio Code*. Dalam *Visual Studio Code*, telah disediakan fitur yang dapat digunakan secara gratis tanpa perlu melakukan instalasi terlebih dahulu.

Performa *Visual Studio Code* yang selalu *ter-update* juga membuat performa *Visual Studio Code* lebih cepat dan lancar. Meskipun memakan lebih

banyak penggunaan memori RAM, ketika pengguna mengalami kesulitan, dapat bertanya ataupun mencari penyelesaian atas masalah serupa yang dialami oleh pengguna lain dengan adanya komunitas yang besar [160], berbeda dengan *sublime text* yang masih minim komunitas penggunanya. Dapat disimpulkan bahwa *Visual Studio Code* menawarkan lebih banyak keuntungan dari segi fitur, segi kemudahan dan segi aksesibilitas jika dibandingkan dengan *sublime text* yang masih memiliki beberapa keterbatasan.

3.6 Kebutuhan Sistem

Dalam penelitian, digunakan beberapa perangkat yaitu perangkat keras dan perangkat lunak (*software*), dan perangkat keras (*hardware*) untuk mendukung proses penelitian. Adapun perangkat keras yang digunakan adalah *laptop Acer Swift X* tahun 2021 dengan spesifikasi sebagai berikut:

1. Processor AMD Ryzen 5 5500U.
2. RAM 16GB 4266 MHz.
3. Graphics AMD Radeon.
4. Operating System (OS) Windows 11.

Adapun berikut merupakan perangkat lunak yang digunakan dalam menjalankan penelitian:

1. Python v3.9.5 sebagai bahasa pemrograman yang digunakan.
2. *Anaconda Jupyter Notebook* sebagai *platform* untuk menguji dan melatih model *Convolutional Neural Network*.
3. *Library NumPy, Pandas, Tensorflow*, sebagai *library* yang digunakan untuk mengolah dataset, melatih dan menguji model.
4. *Kaggle* sebagai *repository* dataset yang digunakan dalam penelitian.
5. *Microsoft Word* sebagai platform dokumentasi hasil penelitian dan penulisan laporan.
6. *Microsoft Visual Studio Code*, sebagai platform untuk melakukan perancangan *website* sebagai luaran dari implementasi model.
7. *XAMPP, PHPMyadmin* sebagai penyimpanan database gambar hasil klasifikasi dan database review pada *website*.