

BAB II

LANDASAN TEORI

2.1 Sampah, Sistem Klasifikasi Gambar, *Computer Vision*, *Image Recognition*, *Deep Learning*, *Website*

2.1.1 Sampah

Sampah berkaitan dengan pola hidup masyarakat dan pertumbuhan penduduk yang meningkat setiap tahun [27]. Sampah ditimbulkan dari aktivitas manusia, sesuai dengan tingkat konsumsi warga terhadap barang atau *material* yang digunakan setiap hari. Indonesia merupakan salah satu negara dengan permasalahan sampah yang tinggi, disebabkan oleh adanya peningkatan produksi sampah oleh masyarakat sebesar satu juta ton sampah per hari [28].

Sesuai dengan riset CNN (*Cable News Network*), yang membuktikan bahwa pengelolaan sampah di negara Indonesia masih belum efektif, tercatat sebesar 24% sampah tidak dikelola, hanya 7% sampah yang berhasil di daur ulang, dan 69% sampah berakhir di TPA (Tempat Pembuangan Akhir). Diketahui bahwa sebesar 40% dari data sampah tersebut merupakan jenis sampah non-organik dengan bagan yang sulit diurai bumi [29]. Oleh karena itu, dibutuhkan pengelolaan sampah yang baik agar sampah tidak menumpuk dan mendatangkan potensi masalah kesehatan yang mengancam masyarakat seperti bibit penyakit [30].

2.1.1.1 Jenis Sampah

Pengelolaan sampah dilakukan demi menekan tingginya angka volume sampah, yang membuktikan bahwa jumlah sampah yang menumpuk sebanding dengan tingkat barang yang dikonsumsi warga setiap harinya. Jenis sampah dapat dibagi menjadi beberapa kategori berdasarkan bahan, tingkat bahaya, tempat atau tipe sampah yang dihasilkan, seperti sampah industri maupun sampah rumah tangga [31].

Jenis-jenis sampah secara umum terbagi menjadi dua jenis, yaitu [32]:

a. **Sampah organik**, sampah ini umumnya berasal dari makhluk hidup. Sampah organik dibagi menjadi dua, yaitu:

- 1) Sampah organik basah, merupakan sampah dengan kandungan air cukup tinggi. Contohnya kulit buah dan sisa sayuran.
- 2) Sampah organik kering, merupakan sampah organik lain dengan kandungan air yang kecil. Contohnya kertas, kayu, ranting dan daun pohon yang kering.

b. **Sampah non-organik**, atau disebut juga sebagai sampah non-organik. Sampah jenis ini biasanya dihasilkan dari bahan yang bukan makhluk hidup, biasanya telah melalui berbagai proses teknologi pengolahan sumber daya alam yang tidak dapat diuraikan oleh alam. Contohnya botol plastik, kaleng, dan tas plastik.

Sampah non-organik yang sulit terurai secara alami oleh alam dapat dikurangi dengan cara mendaur ulang sampah dan mengurangi pemakaian bahan sekali pakai, seperti plastik. Dengan melakukan daur ulang sampah, tidak hanya memberikan dampak positif bagi lingkungan, namun juga bermanfaat bagi kesehatan masyarakat demi terciptanya lingkungan yang bersih [33]. Beberapa contoh sampah organik dan non-organik seperti pada tabel 2.1 [34]

Tabel 2.1 Jenis sampah organik dan non-organik

Jenis Sampah	
Organik	Non-organik
Sisa makanan dari sampah rumah tangga (Buah, sayur, cangkang telur, tulang dan kotoran hewan)	Kantong Plastik
Daun kering dan basah	Ban Bekas
Rumput	Botol Kaca, Botol Plastik, Kaleng
Serbuk kayu, kertas	Tali, benang, wol, baterai

Jenis Sampah	
Sampah rumah tangga	Gelas, kaca, beling, logam

Sesuai dengan pengelompokkan jenis sampah pada tabel 2.1, sampah organik merupakan sampah yang dapat diuraikan dengan mudah melalui proses alami, contohnya daun, ranting, sayuran, buah dan sisa makanan [35], seperti yang ditampilkan pada gambar 2.1.



Gambar 2.1 Contoh sampah organik

Sumber: LPM Patriotik Jambi [35]

Di sisi lain, sampah non-organik, seperti yang tampak pada gambar 2.2 merupakan sampah yang membutuhkan waktu lama untuk terurai, contohnya botol plastik, tas sampah plastik, dan kaleng [36].



Gambar 2.2 Contoh sampah non-organik

Sumber: LPM Patriotik Jambi [35]

2.1.1.2 Upaya pemilahan sampah

Telah diterapkan banyak cara untuk membekali masyarakat dengan pengetahuan untuk daur ulang sampah, seperti program bank sampah dan TPS-3R. TPS-3R (Tempat Penampungan Sementara)-3R (*Reduce, Reuse, Recycle*) merupakan teknologi pencacah sampah untuk mendaur ulang sampah non-organik berupa plastik dan kardus, serta mengolah sampah organik seperti sisa makanan agar menjadi kompos.

Bank sampah bermanfaat agar warga menabung sampahnya untuk kemudian disumbangkan, warga akan mendapatkan buku tabungan dan dapat meminjam uang [37]. Uang yang dipinjam dapat dikembalikan dengan cara memberikan sampah senilai dengan uang yang dipinjam.

Meskipun bank sampah telah berjalan di sejumlah provinsi, masih ada saja masyarakat yang peduli tentang pemilahan sampah maupun sampah di lingkungan, tak lain adalah membuang sampah sembarangan, ataupun membuang sampah pada tempatnya namun tidak sesuai dengan jenisnya. Misalnya, sampah organik seperti roti, dan sampah sisa makanan yang dibuang ke tempat sampah non-organik, dan sebaliknya yang mengakibatkan banyaknya sampah di tempat pembuangan akhir (TPA) yang tidak dipilah dengan baik sehingga muncul bau yang tidak sedap dan menyebabkan masalah kesehatan masyarakat, misalnya penyebaran penyakit melalui hewan lalat atau tikus. Paparan langsung terhadap limbah TPA juga berbahaya karena dapat menyebabkan masalah pernapasan dan penyakit kulit [38].

Tempat sampah dengan jenis organik dan non-organik telah ditemui tersedia di beberapa sudut ruang ataupun di pinggir jalan raya. Warna universal tempat sampah organik biasanya identik dengan warna hijau, karena melambangkan alam dan kehidupan, sehingga tempat sampah hijau menjadi tempat sampah organik sebagai wadah untuk sisa makanan dan barang berbahan alami yang mudah terurai [39]. Pengolahan dari tempat sampah berwarna hijau (organik) ini nantinya adalah mengolahnya kembali menjadi pupuk kompos. Sementara itu, tempat sampah non-organik dilambangkan dengan warna kuning, yang melambangkan wadah tempat sampah non organik seperti plastik bekas, kemasan air mineral, kaleng, logam dan seterusnya. Tempat sampah dengan warna kuning ini akan dimanfaatkan kembali sebagai kerajinan daur ulang atau untuk dilakukan daur ulang di pabrik [40]. Contoh

tempat sampah berwarna hijau (organik) dan berwarna kuning (non-organik) yang ditemukan di gedung kampus Universitas Multimedia Nusantara terlihat pada gambar 2.3.



Gambar 2.3 Tempat Sampah Organik dan Non-organik

Sumber: dokumentasi peneliti

Meskipun pemerintah setempat telah menyediakan tempat sampah dan menerapkan banyak program untuk mendorong keinginan masyarakat untuk melakukan pemilahan sampah secara mandiri, program tersebut masih belum berhasil menjangkau minat masyarakat secara keseluruhan [41]. Oleh karena itu, perlu diberikan sosialisasi lebih kepada masyarakat dengan menggunakan pendekatan yang lebih menarik, untuk meminimalisir pemikiran skeptis masyarakat tentang pemilahan sampah, dengan harapan membuat lingkungan menjadi lebih indah [42].

Diantaranya, dengan menciptakan sebuah sistem klasifikasi gambar untuk menganalisis objek pada gambar dengan menggunakan bantuan teknologi komputer yang disebut sebagai *Computer Vision*. Teknik dasar pendekatan *computer vision* dalam melakukan analisis objek pada gambar adalah dengan cara mengambil gambar melalui fitur kamera pada *handphone* pengguna, ataupun memuat gambar dari galeri pengguna sebagai sumber masukan (*input*) data [43]. Dengan adanya

sistem klasifikasi gambar melalui bantuan *computer vision*, diharapkan masyarakat akan lebih terbuka dan tertarik untuk mempelajari jenis sampah dan lebih tertarik untuk ikut ambil bagian dalam menjaga lingkungan terutama dengan melakukan pemilahan sampah.

2.1.2 Sistem Klasifikasi Gambar

Sistem klasifikasi gambar adalah sebuah sistem yang dirancang untuk mengelompokkan gambar berdasarkan objek yang terdeteksi pada gambar. Beberapa penelitian sebelumnya telah berhasil mengimplementasikan beragam model *machine learning* atau *deep learning* untuk melakukan klasifikasi gambar. Performa klasifikasi gambar sebagian besar bergantung pada sebaik apa proses ekstraksi fitur yang diimplementasikan pada sistem dari gambar yang digunakan [44].

Sistem klasifikasi gambar tidak hanya digunakan dalam aplikasi sederhana, namun juga bermanfaat dalam beberapa aplikasi seperti kota pintar, agar dapat dilakukan pengawasan lewat pengambilan gambar sampah secara visual pada beberapa lokasi dalam kota tersebut, demi *monitoring* terhadap sampah yang lebih baik [45]. Pada dasarnya, klasifikasi merupakan teknik untuk melakukan pengelompokkan satu atau lebih objek yang diteliti ke dalam sebuah label atau kategori yang sudah ditentukan, contohnya seperti melakukan pengelompokkan tipe-tipe sel darah manusia, dan mengelompokkan hewan berdasarkan kelasnya [46].

Klasifikasi dapat dilakukan pada data terstruktur maupun tidak terstruktur, dengan isi dari dataset akan diklasifikasikan berdasarkan label ataupun kategori yang telah ditentukan sebelumnya sebagai input baru untuk melakukan prediksi. Model klasifikasi menyimpulkan fungsi *mapping* (penggambaran hubungan) yang *valid* dari data *training* dan memprediksi label kelas dengan bantuan fungsi *mapping* ketika adanya pemasukan data baru. Label dalam klasifikasi dibedakan menjadi dua tipe, yaitu [47]:

- a) **Binary classification**, merupakan klasifikasi dengan satu ataupun dua keluaran. Contohnya, seperti ramalan cuaca (apakah akan hujan atau tidak), *spam or not spam* (memprediksi apakah sebuah email adalah spam atau bukan), dan diagnosa medis klasifikasi pasien golongan sehat atau sakit.
- b) **Multi-label classification**, klasifikasi dengan lebih dari dua kemungkinan. Contohnya, mengklasifikasikan performa akademik dari seorang siswa apakah termasuk sangat baik (*excellent*), atau baik (*good*), biasa saja (*average*), atau kurang (*poor*).

Teknik klasifikasi gambar yang digunakan dalam penelitian adalah *binary classification*. *Binary classification* merupakan salah satu metode yang digunakan dalam *supervised learning* (dijelaskan lebih lanjut pada sub-bab 2.1.5 *deep learning*). *Binary classification* yang digunakan dalam penelitian ini bertujuan untuk mengklasifikasi gambar sampah, apakah tergolong ke dalam kategori organik ataupun kategori non-organik. Hasil dari klasifikasi *biner* biasanya dilambangkan dengan angka 0 dan 1. Dalam penelitian ini, angka 1 merepresentasikan jenis sampah non-organik dan angka 0 merepresentasikan jenis sampah organik. Hasil *binary classification* dapat dievaluasi dengan menggunakan *confusion matrix*, untuk melihat kebenaran maupun kesalahan prediksi model [48].

Perkembangan konten digital yang terus berkembang memicu munculnya beragam pembelajaran untuk melakukan klasifikasi gambar secara otomatis, dengan bantuan teknologi penglihatan komputer dalam melakukan analisa objek pada gambar dengan model dari algoritma yang dijalankan dibalik sebuah sistem. Bantuan teknologi penglihatan dari komputer dalam melakukan pengenalan objek pada gambar disebut sebagai *computer vision* [49].

2.1.3 *Computer Vision*

Computer vision merupakan sebuah cabang pembelajaran yang berfokus pada penglihatan komputer untuk memahami isi konten dari sebuah

gambar digital. *Computer Vision* menerapkan metode reproduksi kemampuan penglihatan manusia dalam mengenali objek dalam gambar, yang bergantung pada fitur dalam gambar untuk mendeskripsikan poin, ujung (*edges*), objek, maupun warna [50]. *Computer vision* melakukan klasifikasi gambar dengan cara menguraikan gambar menjadi beberapa *pixel*. Setiap *pixel* direpresentasikan dengan nilai yang berbeda. Beberapa tugas (*task*) umum yang dapat dilakukan oleh *computer vision*, yaitu [51]:

- a. ***Image classification***, mengklasifikasikan gambar menjadi beberapa kelas yang telah ditentukan.
- b. ***Object detection***, menentukan dan mengetahui letak objek dari sejumlah kategori yang telah di tentukan pada sebuah gambar alami ataupun objek tertentu yang diberikan.
- c. ***Image segmentation***, membagi gambar menjadi beberapa bagian, seperti mengklasifikasikan setiap *pixel* menjadi entitas tertentu yang spesifik.

Image Classification menerapkan sebuah teknik klasifikasi untuk mengenali gambar, yang disebut *Image Recognition*. *Image Recognition* bekerja dengan cara mengenali objek pada gambar yang kemudian dilanjutkan oleh *computer vision*, untuk melakukan klasifikasi gambar, deteksi objek, maupun segmentasi gambar [52].

2.1.4 ***Image Recognition***

Image Recognition merupakan sebuah ranah pembelajaran *computer vision* (penglihatan komputer) yang dapat melakukan identifikasi dan klasifikasi objek dalam gambar digital. Tujuan dari *Image recognition* adalah membagi kategori gambar menjadi beberapa bagian, agar ketika gambar diunggah, akan menampilkan sebuah informasi tentang apa saja objek yang ada dalam gambar tersebut [53].

Dengan perkembangan teknologi komputer, teknologi gambar ini tidak dapat dipisahkan dari teori *neural networks*, sebagai dan kecerdasan buatan (*artificial intelligence*). Manfaat dari melakukan pemrosesan gambar yaitu

untuk menentukan apakah gambar sudah mengeluarkan *output* dengan informasi dapat dibaca. Cara kerja *Image recognition* hampir mirip dengan cara manusia melihat gambar dan membaca informasinya. Teknologi *image recognition* dibagi menjadi beberapa tahapan, yaitu sebagai berikut [54]:

- a. ***Information acquisition***, merupakan tahapan mencari informasi awal dari objek yang hendak diteliti dan mengubahnya menjadi informasi yang dapat dikenali oleh mesin dengan beberapa arti yang bermanfaat.
- b. ***Data preprocessing***, merupakan teknik untuk transformasi dalam pemrosesan gambar, seperti meningkatkan fitur penting dari gambar (*image*), sebelum mulai masuk ke tahapan pemrosesan data. Umumnya, *data preprocessing* mencakup beberapa tahapan utama seperti pembersihan data, transformasi data, dan pengurangan data [55]. Sedangkan untuk klasifikasi gambar, secara umum dilakukan *grayscale*, *thresholding*, dan *image segmentation* [56]. Namun, tahapan *data preprocessing* yang ditempuh dalam penelitian ini hanya mencakup proses *image rescaling*. Alasannya, karena fokus penelitian adalah klasifikasi gambar dan informasi warna tidak memiliki pengaruh yang relevan dalam klasifikasi gambar. Proses *grayscale* dan *thresholding* hanya dilakukan pada penelitian yang berfokus pada deteksi objek spesifik pada gambar, sehingga diperlukan penyetaraan warna pada *background* dan latar belakang [57]. Berikut merupakan penjelasan mengenai *Image Resizing* [58].

1. ***Image Resizing (Image Rescaling)***

Image Resizing, disebut juga proses *resize*, bermanfaat untuk mengubah ukuran dari sebuah gambar menjadi lebih kecil maupun lebih besar, agar dapat mencapai ukuran yang diinginkan. Manfaat dilakukan *resize* selain untuk memperbesar maupun memperkecil gambar, yaitu untuk memperbaiki kualitas gambar. Selain itu, tidak menutup kemungkinan hasil dari *resizing* dapat memperburuk kualitas gambar

apabila gambar memiliki resolusi rendah [58]. Dalam penelitian ini, nilai *input* gambar diatur menjadi hingga berukuran 224x224 *pixel*.

- c. ***Feature extraction and selection***, proses ini merupakan proses pencarian fitur dalam gambar, mirip dengan proses mengenali pola dalam sebuah gambar, seperti saat melakukan identifikasi karakteristik gambar. Tahapan ini merupakan fokus utama yang penting dalam *image recognition*, dalam menemukan fitur penting yang dapat di ekstraksi.
- d. ***Classifier design and classification decision***, tahapan hasil ekstraksi fitur kemudian diklasifikasikan berdasarkan *classifier* pilihan dari model algoritma yang digunakan dalam penelitian. Apakah model yang dirancang akan meneliti pola, ataupun mengenai objek pada gambar.

Image recognition merupakan hasil gabungan dari kecerdasan buatan (*artificial intelligence*) dan penglihatan komputer (*computer vision*). *Image recognition* sering digunakan dalam pembelajaran *deep learning*, sebagai ranah dilangsungkannya pengembangan sistem untuk melangsungkan penelitian dan melakukan analisis gambar [59].

2.1.5 *Deep Learning*

Deep Learning merupakan bagian dari kecerdasan buatan (*artificial intelligence*), yang dapat membantu manusia dalam memecahkan masalah yang sangat sulit untuk diselesaikan. Kelebihan kecerdasan buatan yaitu dapat menerapkan pola pikir dan belajar seperti halnya cara otak manusia berpikir. *Artificial Intelligence* dibagi menjadi dua pembelajaran, yaitu *deep learning* dan *machine learning* [60]. Dalam penelitian ini, cabang pembelajaran yang digunakan adalah *deep learning*.

Dalam *deep learning*, performa algoritma meningkat seiring bertambahnya *volume data*. Berbeda dengan *machine learning* yang bergantung pada fitur aplikasi dari keputusan buatan (*artificial determination*), *deep learning* lebih berfokus pada percobaan dengan level yang tinggi untuk mendalami fitur dari data dan mendapatkan model *machine learning* yang lebih dalam (*deep*) melalui beragam fitur hasil transformasi [61].

Fitur utama sebagai bahan pembelajaran mendalam (*deep learning*) adalah *data train* (data latih), yang digunakan untuk melatih model. Jumlah data *train* biasanya berjumlah lebih besar dibandingkan *data test*. Setelah mendapatkan data *training*, mesin akan melakukan pembelajaran terhadap data dan melakukan teknik klasifikasi pola untuk mengenali objek pada gambar [62]. Klasifikasi tersebut dapat dibagi menjadi dua jenis berdasarkan teknik pembelajarannya, yaitu [63]:

- a) ***Supervised learning***, proses klasifikasi dengan menggunakan data dan label yang telah diketahui untuk melakukan klasifikasi pada data asing. Contoh algoritma yang sering digunakan adalah: *Logistic Regression*, *Naïve Bayes*, *Support Vector Machine*, *Artificial Neural Network*, *Convolutional Neural Network*, dan *Random Forests*.
- b) ***Unsupervised learning***, proses klasifikasi dengan data dan label yang tidak diketahui sebelumnya sehingga perlu melalui proses *clustering* terlebih dahulu sebelum melakukan klasifikasi, dengan tujuan membagi data asing menjadi beberapa kelompok dalam jumlah tertentu berdasarkan fiturnya. Contoh algoritma yang sering digunakan adalah *K-Means Clustering*, *PCA (Principal Component Analysis)*, dan *Autoencoder*.

Pemilihan penggunaan *supervised learning* maupun *unsupervised learning* dilakukan sesuai dengan data yang dimiliki dan menyesuaikan dengan kebutuhan dalam sebuah penelitian. Dalam penelitian ini, teknik klasifikasi yang digunakan adalah *supervised learning*. Keuntungan menggunakan *supervised learning* yaitu model dapat memperoleh keahlian yang baik untuk memprediksi secara terstruktur dan akurat. Namun, kelemahannya yaitu saat *training* model dibutuhkan waktu yang cukup lama, dan pembelajaran algoritma bergantung pada isi *dataset* yang digunakan [64]. Algoritma *supervised learning* yang paling sering digunakan untuk metode analisis dan klasifikasi gambar yaitu algoritma *Convolutional Neural Network (CNN)* [65].

Dalam melakukan klasifikasi gambar dengan metode *supervised learning*, CNN termasuk algoritma dengan performa yang sangat baik dalam melakukan klasifikasi gambar [66]. Meskipun demikian, CNN memiliki beberapa kelemahan, yaitu waktu *training model* yang lama, dan biaya komputasi yang tinggi (dalam mempercepat *training* perlu menggunakan GPU) [67]. CNN juga dipengaruhi oleh beberapa tantangan untuk mencapai parameter yang jumlahnya jutaan, seperti kurangnya dataset *training* yang dapat dipercaya oleh model, sehingga muncul beberapa kasus pada algoritma CNN yang mengalami *overfitting* [68].

Overfitting merupakan sebuah keadaan munculnya *error* yang sedikit pada saat model dilatih dengan menggunakan data latih (*data train*), namun akan muncul nilai *error* yang tinggi ketika diujikan pada data uji (*data test*) [69]. Permasalahan *overfitting* dapat diatasi dengan melakukan regularisasi (pembiasaan) terhadap data. Contohnya dengan menambahkan *training* dalam transformasi seperti *scaling*, *zooming*, *early-stopping*, serta melakukan pemotongan gambar secara acak [70]. Dalam penelitian ini, metode pencegahan *overfitting* yang digunakan untuk meminimalisir munculnya *error* pada data yang diuji nantinya adalah dengan metode *early-stopping*.

Berbeda dengan *machine learning* yang hanya berfokus pada prediksi dan pembuat keputusan tanpa melalui pemograman yang kompleks, *deep learning* menekankan kedalaman struktur model yang biasanya terdiri dari dua hingga enam, ataupun lebih dari enam *hidden layer* dalam sebuah *neural networks*. *Deep learning* dapat mempelajari fitur dari data secara otomatis, serta memudahkan pembelajaran dan pengaturan parameter dalam studi *big data* dan cocok dalam memecahkan permasalahan kompleks seperti lapisan transformasi *non-linear* [71].

Dalam ranah *image recognition*, mengekstraksi fitur dalam gambar menjadi bagian paling kritis dalam sistem pengenalan objek atau pola. Melalui konversi *layer demi layer*, *deep learning* akan mendapatkan fitur ekspresi yang paling baik [72]. Hasil *output* dari implementasi model algoritma dalam *deep*

learning dapat berupa *mobile* maupun *website*, pada penelitian ini dipilih *website* sebagai hasil *output* implementasi model algoritma CNN. Adapun alasan pemilihan *website* akan dijelaskan pada sub-bab 2.1.6.

2.1.6 Website (Situs Web)

Arti dari *Website* (situs *web*) ialah sebuah halaman *web* yang memuat kumpulan informasi yang terhubung antar pribadi, kelompok, maupun organisasi. Situs web yang dapat tergolong baik jika mampu menampilkan visual yang menarik dan bermanfaat bagi penggunanya, berdasarkan pemahaman Rochmawati (2019) [73]. *Website* untuk digunakan sebagai wadah untuk mengimplementasikan algoritma dalam sistem klasifikasi gambar, karena kompatibilitas dalam penggunaan, yaitu dapat diakses melalui *handphone* dengan sistem operasi *Android* maupun *iOS*, melalui PC ataupun laptop, selama terhubung dengan jaringan koneksi internet [74].

Menurut Abdulloh (2018), *website* merupakan halaman dengan kumpulan informasi data digital berupa teks, gambar, animasi, suara, video, ataupun gabungan dari semuanya. Pada dasarnya, semua informasi yang disajikan melalui jalur koneksi internet dapat diakses dan dilihat oleh semua orang di seluruh dunia. Sebagai bahasa standar untuk menampilkan konten pada halaman *website*, digunakan HTML (*Hyper Text Markup Language*) [75]. *Website* berfungsi untuk menampilkan dan menguji hasil program. Menurut penjabaran Hasugian, fungsi dari *website* adalah sebagai berikut [76]:

- 1) Sebagai media pemasaran
- 2) Sebagai media informasi
- 3) Sebagai media komunikasi
- 4) Sebagai media promosi
- 5) Sebagai media pendidikan

Dalam merancang situs *web*, pada umumnya digunakan 3 bahasa pemrograman yang umum digunakan, yaitu:

- 1) **HTML (*Hyper Text Markup Language*)**

Bahasa pemrograman khusus pengembangan halaman *web* untuk melakukan organisir dan format dokumen [77].

2) **CSS (*Cascading StyleSheet*)**

Bahasa pemrograman yang berfungsi untuk melakukan desain pada tampilan sebuah *web*, seperti melakukan konfigurasi warna, *font*, *background*, dan menyesuaikan tampilan *web* dengan layar, dan lain sebagainya [78]. CSS digunakan bersamaan dengan HTML agar tampilan *website* menjadi lebih menarik.

3) **PHP (*Personal Home Page*)**

Bahasa pemrograman berbasis *script* untuk melakukan pemrograman dalam situs *web*, seperti melakukan koneksi ke *database*, untuk menampilkan *output* pada situs *web* pada pihak klien. PHP dapat membaca, menulis file maupun visualisasi gambar atau *video* [79].

Selain *Website*, pilihan aplikasi yang dapat digunakan sebagai luaran dari penelitian tak lain adalah *Android*. Oleh karena itu, pada tabel 2.2 akan dibahas mengenai perbandingan *Website* dengan *Android*:

Tabel 2.2 Perbandingan Website dan Android

No.	Indikator Pemanding	Website	Android
1	Aksesibilitas platform	Komputer, laptop, tablet, <i>smartphone iOS</i> dan <i>Android</i>	<i>Smartphone Android</i>
2	Kemudahan pengembangan dan pemeliharaan	Dapat diakses di <i>platform</i> manapun dengan versi yang sama	Perlu membangun versi yang berbeda tiap <i>platform</i>
3	Instalasi aplikasi tambahan untuk mengakses informasi atau layanan	Tidak perlu instalasi	Memerlukan instalasi
4	Fleksibilitas	Langsung	Memerlukan persetujuan pengguna

No.	Indikator Pemanding	Website	Android
5	Aksesibilitas melalui mesin pencari	Dapat dioptimalkan agar mudah dicari secara <i>online</i>	Tidak dapat diakses
6	Biaya pengembangan	Ekonomis karena dapat diakses melalui berbagai <i>platform</i>	Membutuhkan biaya yang berbeda untuk setiap <i>platform</i>

Berdasarkan tabel 2.2, terlihat bahwa *website* memiliki keunggulan yang lebih dominan dibandingkan dengan *android*, terutama kemudahan untuk pemeliharannya, dan aksesibilitas lintas platform sehingga *website* dapat diakses kapan dan dimana saja [80], oleh karena itu, *website* dipilih sebagai luaran dari penelitian ini.

2.2 Framework dan Algoritma

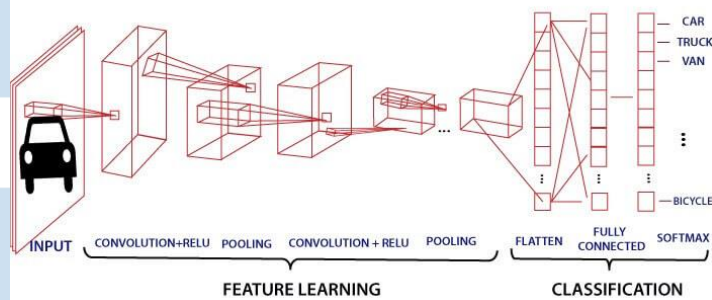
Berikut merupakan model algoritma dan *framework* yang digunakan dalam proses pembuatan sistem klasifikasi gambar berbasis *website*, diantaranya:

2.2.1 Algoritma CNN (*Convolutional Neural Network*)

CNN (*Convolutional Neural Network*) merupakan algoritma klasifikasi dengan metode *supervised learning* dalam pembelajaran *deep learning*, yang berbasis *image recognition* dan *computer vision*. CNN pertama kali dikembangkan dengan sebutan *NeoCognitron* oleh Fukushima pada tahun 1980, merupakan *network multi-layer* yang bekerja dengan cara mengenali pola visual yang kuat melalui pembelajaran [81]. CNN merupakan pengembangan dari MLP (*Multi-Layer Perceptron*) yang dapat digunakan untuk mendeteksi dan mengenali data dua dimensi [82].

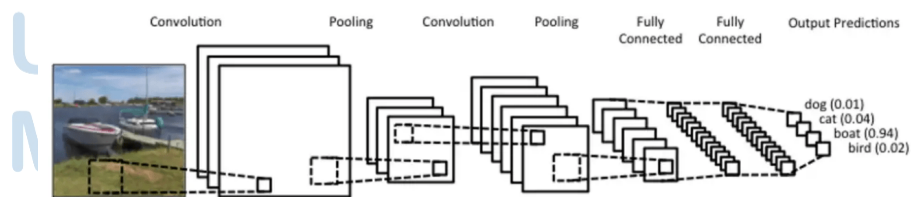
Cara kerja CNN sebenarnya mirip dengan MLP, hanya saja CNN merepresentasikan setiap neuron dalam dua dimensi, sedangkan MLP hanya satu dimensi. CNN melakukan klasifikasi terhadap data yang telah memiliki

label sebelumnya (*supervised learning*) dengan cara melatih data (*data train*) dan menargetkan *variable*, sehingga suatu data dapat dikelompokkan ke kategori yang sudah ada. Pada gambar 2.4, merupakan gambaran arsitektur CNN secara umum yang terdiri dari dua bagian, yaitu *feature learning* dan *classification*.



Gambar 2.4 Gambaran arsitektur CNN
Sumber: [83]

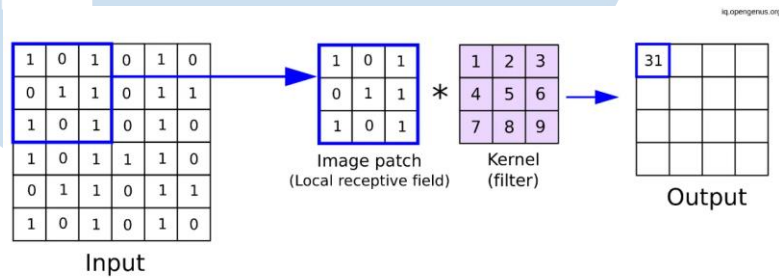
Bagian *feature learning* mengubah gambar yang di *input* menjadi fitur yang dikonversikan dalam bentuk angka sebagai bentuk informasi yang merepresentasikan gambar sebelum dapat dilanjutkan ke proses *classification*. *Feature learning* terdiri dari *layer convolutional* dan *layer pooling* yang menerima *input* dari gambar, kemudian diproses dan menghasilkan *output* berupa *array* multi dimensi. Sementara itu, bagian *classification* merupakan *layer* dengan neuron yang memiliki koneksi secara penuh (*fully connected layer*), yang menerima *input* dari *map* fitur pada *layer* sebelumnya lalu kemudian dilanjutkan ke proses *flatten* yang akan mengubah dimensi dari data yang didapat menjadi bentuk vektor tunggal agar dapat diproses *hidden layer* dan beberapa *fully connected layer* selanjutnya, untuk menghasilkan luaran berupa tingkat akurasi dari hasil klasifikasi setiap kategori atau kelas [84].



Gambar 2.5 Gambaran model CNN
Sumber: [85]

CNN mengkombinasikan beberapa *layer* dalam proses filterisasi saat melakukan *training* pada model, seperti yang terlihat pada gambar 2.5, model CNN terdiri dari *convolutional layer*, *pooling layer*, dan *fully connected layer*. *Layer-layer* yang digunakan dalam merancang model CNN dapat dijabarkan sebagai berikut [86]:

- a. *Convolutional layer*, bagian proses utama CNN, dengan adanya operasi konvolusi yang mengaplikasikan perhitungan setiap *filter* menuju bagian *pixel* lain sesuai dengan ukuran *kernel* yang digunakan. Contohnya, pada gambar 2.6 setiap *filter* mengalami pergeseran dalam perhitungan antara nilai *input* (*pixel* gambar) dan nilai *filter*, yang dikalikan untuk kemudian menghasilkan sebuah nilai *output*. Operasi ini mirip dengan operasi perkalian matriks.

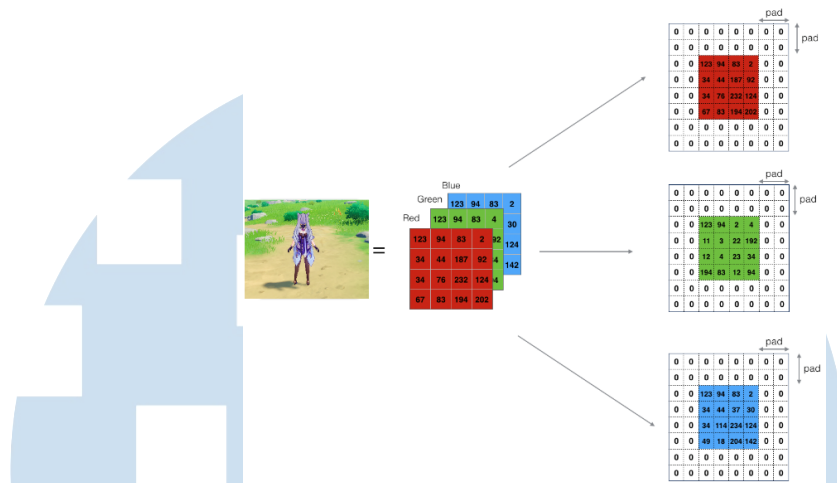


Gambar 2.6 Proses *convolutional layer*

Sumber: [87]

Proses tersebut akan menghasilkan sebuah *feature map* berbentuk dua dimensi, dengan panjang, lebar, dan tinggi (pixel) yang sesuai dengan jumlah *channel* warna dari *input*. *Channel* warna yang dimaksud adalah warna RGB (*red, green, blue*), seperti contoh pada gambar 2.7.

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 2.7 Contoh gambar RGB

Sumber: [88]

Persamaan dari operasi konvolusi CNN dapat dituliskan seperti pada persamaan (1) [89]:

Rumus 1.1 Persamaan operasi konvolusi CNN

$$s(t) = (x * w)(t) \tag{1}$$

Keterangan:

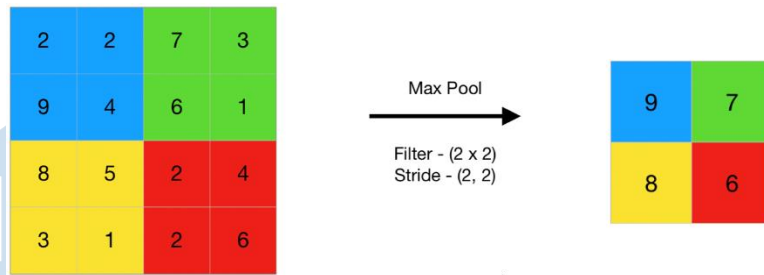
$s(t)$ = Fungsi hasil operasi konvolusi

x = *Input*

t = *pixel*

Fungsi $s(t)$ pada rumus (1) menghasilkan *output* berupa *feature map*, *input* pertama adalah x , dan w berperan sebagai *kernel* atau *filter*, kemudian t sebagai *pixel*.

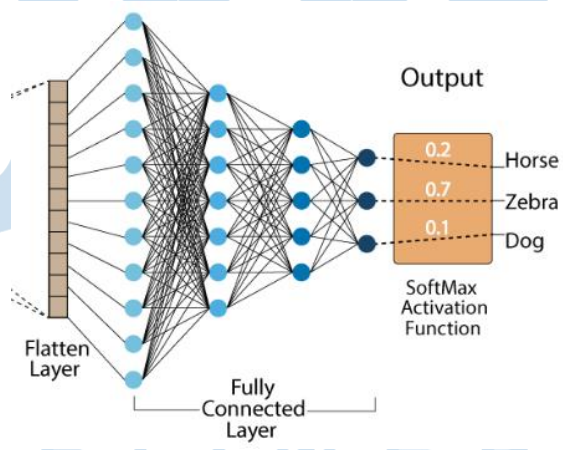
- b. *Pooling layer*, yaitu *layer* berisi nilai *output* yang didapat dari tahapan *layer* konvolusi yang digunakan. Pada *layer* ini digunakan ukuran *stride* untuk menjaga *filter* yang bergeser pada seluruh area *pixel* dari *feature map*. *Pooling layer* umumnya terbagi atas dua jenis *pooling layer*, yaitu *max pooling* (menggabungkan *pixel* terbesar dari setiap *kernel pooling*) dan *average pooling* (menggunakan nilai rata-rata dari tiap *kernel*).



Gambar 2.8 Contoh proses *max pooling*
 Sumber: [90]

Seperti yang terlihat pada contoh proses *max pooling* pada gambar 2.8, bahwa *max pooling* tersebut memiliki nilai input dengan ukuran 4x4, yaitu masing-masing dari 4 angka pada setiap kotak dengan warna berbeda yang diambil angka maksimalnya untuk dijadikan sebagai *output* dengan ukuran *kernel filter* 2x2 dan nilai *stride* 2.

- c. *Fully connected layer*, merupakan *layer* yang menggunakan hasil *feature map* yang didapatkan dalam proses *pooling* sebelumnya. Namun, sebelum masuk ke *layer fully connected*, *feature map* harus melalui proses *flatten* (meratakan atau *reshaping*), untuk mengubah bentuk array multi-dimensi menjadi sebuah vektor tunggal (satu dimensi) [91]. Letak dari *fully connected layer* yaitu setelah melalui *layer flatten*, seperti pada gambar 2.9.

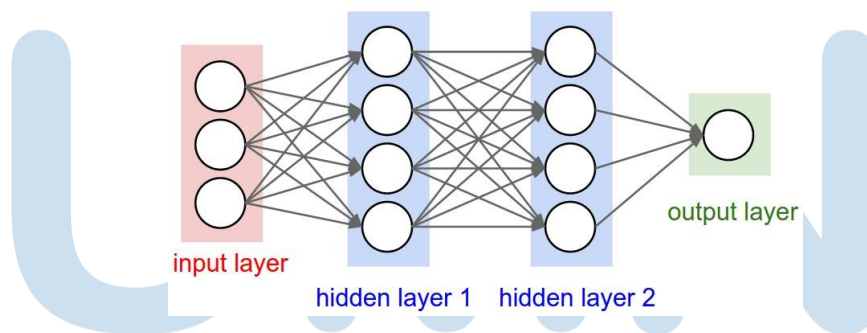


Gambar 2.9 Letak *Fully connected layer*
 Sumber: [93]

Layer flatten sebelum *fully connected layer* berfungsi sebagai *layer* yang digunakan sebelum fitur yang diekstraksi dimasukkan ke dalam *layer dense* dalam *fully connected layers*. Ketika *layer dense* mengambil *input* satu dimensi, *layer flatten* akan meratakan fitur data dan memberikan *output* berbentuk vector tunggal satu dimensi agar dapat diterima oleh *layer dense* [92].

Hasil dari proses *flatten* kemudian akan menjadi *input fully connected layer* atau *dense layer*. *Fully connected layer* terdiri dari beberapa *hidden layer* (contoh *hidden layer* seperti *convolutional layer*, *pooling layer*, atau *dense layer*), dapat juga berupa fungsi aktivasi, langsung menuju *layer output*, ataupun *loss function* seperti pada gambar 2.10. *Hidden layer* bermanfaat untuk merepresentasikan data dengan tingkatan tinggi dan kompleks, dan melakukan transformasi *non-linear* pada *input* [94].

Layer input merupakan *layer* tempat *neural network* menerima masukan dan mengubahnya menjadi beberapa *hidden layer*. *Hidden layer* terdiri dari beberapa neuron yang terhubung ke semua neuron pada *layer* sebelumnya [95].



Gambar 2.10 Lapisan dalam *fully connected layer*
Sumber: [91]

Jumlah *hidden layer* yang optimal untuk dataset skala kecil dan menengah yaitu dapat dimulai dengan satu hingga dua *hidden layers* [96]. Jumlah *hidden layer* dapat ditambah ataupun dikurangi, tergantung pada kompleksitas masalah yang dihadapi dan ukuran dataset yang dimiliki, serta performa model yang diinginkan. Jumlah

hidden layer dapat ditambahkan apabila *task* yang dilakukan merupakan *task* yang sangat kompleks dengan jumlah dataset yang besar dan variasi yang kompleks. Contoh *task* yang tergolong kompleks adalah deteksi objek, pengenalan aksi dalam *video*, *image-captioning* (perpaduan visual + *Nature Language Processing (NLP)*) [97]. Jumlah *hidden layer* dapat ditambah jika diperlukan, dengan cara memonitor performa model melalui akurasi data *validation* terlebih dahulu [98].

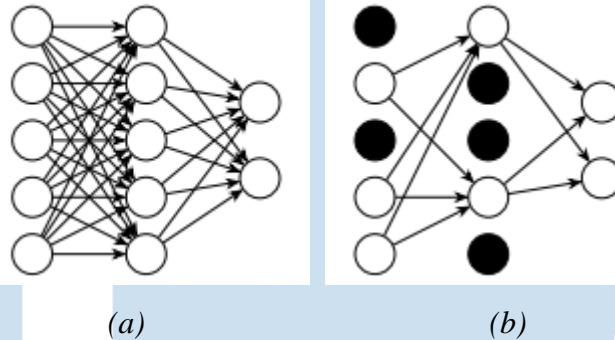
Ketika sebuah *neural network* memiliki jumlah *hidden layer* yang banyak atau mengalami penambahan, maka waktu yang dibutuhkan untuk memproses data pada saat melakukan *training* data juga relatif lama karena model memiliki kompleksitas yang tinggi [99]. Selain waktu yang lama, model juga rentan mengalami *overfitting*, sehingga memerlukan waktu lebih untuk melanjutkan *training* dan perlu ditambah teknik regularisasi (kalibrasi model untuk mengurangi *adjusted loss function* untuk mencegah *overfitting* atau *underfitting*) [100] tambahan.

Dengan menggunakan *hidden layer* yang sedikit atau mengurangi jumlah *layer* dalam *hidden layer*, maka selain menyederhanakan kompleksitas model, komputasi dan waktu untuk *training* data juga menjadi lebih cepat sehingga dapat mencegah terjadinya *overfitting* [101]. Di sisi lain, *layer dense* (atau *fully connected layers*) merupakan *layer* tempat klasifikasi gambar dilakukan. *Layer dense* dalam *fully connected layer* mengubah nilai linier pada *input* dengan menggunakan fungsi aktivasi untuk menghasilkan *output* yang sesuai dengan jumlah kelas dalam klasifikasi [96].

CNN menggunakan pixel orisinal dari gambar *input* secara langsung dan mengaitkannya dengan *filter* ataupun *kernel* untuk melakukan ekstraksi fitur, hal ini membuat akurasi klasifikasi gambar

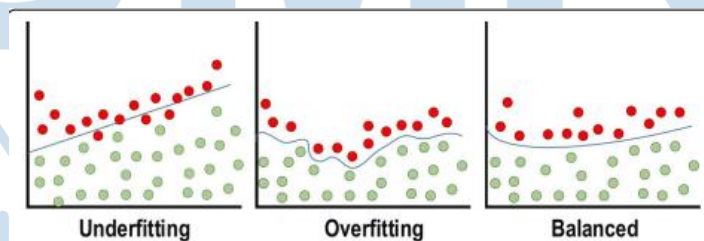
meningkat sehingga model CNN dinilai memiliki performa yang baik [102].

d. *Dropout*



Gambar 2.11 Layer (a) *fully connected* (b) setelah melalui proses *dropout*
Sumber: [103]

Dropout merupakan fungsi yang digunakan untuk mencegah munculnya *overfitting* pada saat proses pelatihan (*training*) model. *Dropout* mengatur *output* dari *neuron* yang tidak terpakai sesuai dengan nilai probabilitas yang ditentukan, contohnya 0.2, maka dalam proses *training*, sebanyak 20% dari *neuron* dalam *layer* tertentu akan dibuang (*dropped-out*) secara acak ataupun di *set* menjadi nol. Pemilihan *neuron* ini dapat menghasilkan hasil yang beragam setiap iterasinya. Cara kerja proses *dropout* terlihat pada gambar 2.11, dimana *network* sebelumnya *layer fully connected* seperti pada gambar (a), melalui proses *dropout* seperti pada gambar 2.11 (b) dengan probabilitas 0.5, sehingga beberapa *neuron* yang tidak digunakan dibuang secara acak [103].



Gambar 2.12 Kondisi *underfitting*, *overfitting*, dan *balanced*
Sumber: [104]

Overfitting merupakan kondisi data setelah melalui tahapan *training* dan mendapatkan akurasi yang baik namun tidak sesuai

dengan hasil prediksi pada data *test* maupun data yang belum pernah dilihat sebelumnya. Contohnya ketika didapatkan nilai akurasi yang tinggi, namun nilai akurasi *validation* yang rendah, mengakibatkan model memiliki kompleksitas tinggi kemudian mempelajari terlalu banyak hal selain *relationships* ataupun pola dalam data *training* [104].

Selain *overfitting*, model juga berpotensi mengalami *underfitting*, dimana model kurang memiliki kapasitas dalam mempelajari kompleksitas dalam data, sehingga performa model kurang maksimal dalam mengenali data *training*, maupun ketika diujikan pada data *testing*. *Underfitting* terjadi ketika model gagal dalam menangkap pola tertentu dalam data *training*. Adapun gambaran dari kondisi *overfitting* seperti yang dapat dilihat pada gambar 2.12 [104].

Kondisi terbaik dalam melatih model adalah akurasi yang stabil (*balanced*), sehingga akurasi yang didapatkan sesuai dengan hasil prediksi pada data baru yang belum pernah dilihat model, dan nilai akurasi dan akurasi *validation* yang setara. Terdapat beberapa fungsi yang sering digunakan dalam membuat model *convolutional neural network*, penjelasannya masing-masing antara lain sebagai berikut.

2.2.1.1 Activation Function

Activation function (fungsi aktivasi), disebut juga fungsi *transfer* merupakan fungsi yang digunakan dalam *neural network* untuk menghitung jumlah *weight* dari *input* dan *biases*, untuk menentukan apakah sebuah neuron dapat digunakan atau tidak.

Fungsi aktivasi mengendalikan *output* dari *neural network* yang mencakup parameter dalam data tersebut. Fungsi aktivasi dapat berbentuk linear maupun *non-linear*, tergantung dari fungsi yang digunakan [105]. Jenis-jenis fungsi aktivasi yang umum

digunakan pada model *neural network*, diantaranya sebagai berikut:

1. *ReLU*

ReLU (Rectified Linear Unit) activation function merupakan fungsi aktivasi yang sering digunakan dalam aplikasi *deep learning*. *ReLU* merepresentasikan fungsi dan model linear yang mudah dioptimalkan [96]. Fungsi *ReLU* ketika mengoperasikan *threshold* pada setiap elemen *input* yang bernilai kurang dari nol, maka nilai tersebut akan dihitung sebagai nol, seperti pada persamaan (2) [105]:

Rumus 1.2 Persamaan fungsi aktivasi ReLU

$$ReLU(x) = \max(0, x) \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases} \quad (2)$$

Persamaan (2) memperbaiki nilai dari input yang bernilai kurang dari nol, dengan mengganti nilainya menjadi nol dan mengeliminasi permasalahan *gradient* yang mungkin hilang, seperti yang dialami dalam fungsi aktivasi yang lain.

2. *Sigmoid*

Sigmoid, disebut juga sebagai *logistic activation function*, merupakan fungsi aktivasi *non-linear* yang digunakan pada nilai *real* sebagai *input* dan *output*, dengan rentang nilai 0 hingga 1 [106]. *Output* dari setiap unit tertindih, sehingga *gradient* akan perlahan akan hilang. Persamaan *sigmoid* yang digunakan ketika mengambil nilai *real* yang dimaksud seperti pada persamaan (3) [107]:

Rumus 1.3 Persamaan fungsi aktivasi Sigmoid

$$\sigma(x) = \left(\frac{1}{1+e^{-x}} \right) \quad (3)$$

Dengan σ sebagai fungsi *sigmoid*, x sebagai nilai *input*, dan e sebagai dasar nilai logaritma (umumnya bernilai 2.71828). *Sigmoid* merupakan *function* yang muncul pada *layer output*

dalam arsitektur *deep learning*. *Sigmoid* sering digunakan dalam memprediksi probabilitas berdasarkan *output* dan diimplementasikan dalam permasalahan *binary classification* dan *modelling logistic regression*, seperti halnya *task neural network* lainnya [108].

2.2.1.2 *Optimizer*

Optimizer (fungsi optimasi) berfungsi untuk menentukan cara parameter model diatur dalam proses *training* untuk meminimalisir perbedaan antara *output* prediksi dengan *true labels* (*loss function*), dengan cara memperbaharui parameter berdasarkan *gradients* komputer secara berulang. *Optimizer* mengarahkan proses *training* dan membantu model dalam menemukan set parameter yang paling optimal untuk performa terbaik dalam peningkatan akurasi [109]. *Optimizer* yang umum digunakan dalam meningkatkan akurasi adalah *RMSprop*, *SGD*, dan *Adam*, yang dapat dijabarkan sebagai berikut:

a. *RMSprop*

RMSprop merupakan *optimizer* yang sering digunakan dalam *machine learning* dan *deep learning*. *RMSProp* (*Root Mean Square Propagation*) membagi gradien dengan akar kuadrat rata-rata (*Root Mean Square* (*RMS*)) [110]. Teknik *learning rate* yang digunakan adalah adaptif, yaitu *learning rate* dibagi dengan rata-rata nilai gradien menjadi gradien kuadrat yang hilang secara eksponensial (*exponentially weighted moving average*). Pembagian gradien ini membuat proses *training* menjadi lebih stabil [111].

RMSprop mempunyai nilai standar untuk *learning rate* sebesar 0,001. Tahapan dalam mengakumulasi gradien kuadrat (*accumulated squared gradient*) yaitu dengan cara menghitung gradien seperti pada persamaan (4) [110].

Rumus 1.4 Persamaan *optimizer RMSprop*

$$r = \rho r + (1 - \rho)g * g \quad (4)$$

$$\Delta\theta = -\frac{\alpha}{\sigma + \sqrt{r}} * g \quad (5)$$

$$\theta = \theta + \Delta\theta \quad (6)$$

Pembaruan parameter (*update parameter*) dihitung dengan menggunakan persamaan (5) dan (6). Dengan g adalah gradien parameter, r sebagai *accumulated squared gradient*, ρ adalah *decay rate*, $\Delta\theta$ adalah *update parameter*, α merupakan *learning rate*, dan θ merupakan parameter awal [112].

b. SGD

SGD (*Stochastic Gradient Descent*) merupakan sebuah algoritma optimasi yang digunakan untuk *training* neural networks. SGD secara iteratif memperbaharui parameter model dalam arah *gradien negative* [113]. Kata *stochastic* dalam SGD berarti bagian dari random *subset (mini-batch)* dari sample *training* yang digunakan untuk mengestimasi *gradien*. SGD mempunyai nilai standar untuk *learning rate* sebesar 0,001.

Pemrosesan dalam setiap *epoch* pada *neural networks* memiliki tujuan untuk mengurangi *error*, yang biasa disebut *cross entropy*. Variabel akan diperbaharui dengan menggunakan time step variable sebelumnya untuk dikurangi dengan *result* dari *learning rate* dikali dengan *gradient vector*, seperti pada persamaan (7) [114].

Rumus 1.5 Persamaan *optimizer SGD*

$$\omega_{(n+1)} = \omega_{(n)} - Lr * \hat{g} \quad (7)$$

Dengan nilai $\omega_{(n)}$ sebagai variabel pembaharuan pada *time n*, Lr sebagai *Learning Rate*, dan \hat{g} sebagai gradien vektor.

c. Adam

Adam (*Adaptive Movement Estimation*) merupakan algoritma optimasi yang berasal dari perluasan SGD yang menggabungkan *learning rates* adaptif. Adam mengkalkulasi *learning rates* untuk setiap parameter menggunakan estimasi dari moment pertama dan kedua dari gradien, yang diperbaharui berdasarkan statistik gradien [114]. Adam mempunyai nilai standar untuk *learning rate* sebesar 0,001. Pada setiap time step, optimizer Adam akan melakukan performa seperti dengan menjalankan persamaan (8), (9), dan (10) [114].

Rumus 1.6 Persamaan optimizer Adam

$$\omega_{(n+1)} = \omega_{(n)} - \frac{L_r}{\sqrt{\hat{v}_{n+\varepsilon}}} \hat{m}_n \quad (8)$$

dan

$$\hat{m}_n = \frac{\beta_1 m_{n-1} + (1-\beta_1) \hat{g}_n}{1-\beta_1^n} \quad (9)$$

$$\hat{v}_n = \frac{\beta_2 v_{n-1} + (1-\beta_2) \hat{g}_n^2}{1-\beta_2^n} \quad (10)$$

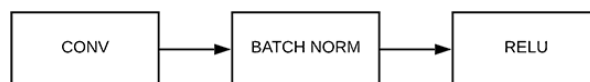
Dengan nilai $\omega_{(n)}$ sebagai variabel pembaharuan pada *time n*, L_r sebagai *Learning Rate*, \hat{g} sebagai gradien vector, \hat{m}_n sebagai estimasi dari *first moment* (mean) dari *gradients*, dan \hat{v}_n sebagai *second moment* (*uncentered variance*) dari *gradients*.

2.2.2 Tensorflow

Tensorflow adalah sebuah *library* dalam *deep learning* yang bersifat fleksibel dan memiliki banyak APIs seperti *Keras* dan *Estimators* dengan beragam modul sederhana, antarmuka konsisten, dan pesan *error* yang jelas, memudahkan penggunaannya untuk dalam melakukan *training*, *compile* model, dan visualisasi model seperti *neural network* dan model *deep* atau *machine learning* lainnya [115].

2.2.3 Model *Sequential Keras*

Model *Sequential* merupakan bagian dari API library Tensorflow Keras yang paling sederhana dan umum digunakan dalam membuat model *deep learning*. Model *Sequential* didefinisikan sebagai model dengan sekuens (rangkaian) *layer*, dimana peneliti dapat mengatur model secara *layer* demi *layer* untuk membangun sebuah *neural networks* [116].



Gambar 2.13 Model *Sequential Keras*

Sumber: [116]

Seperti contoh pada gambar 2.13, model *Sequential* dibuat terlebih dahulu baru kemudian ditambahkan *layer*, diawali dengan *layer* input, kemudian layer 1 (konvolusi), layer 2 (konvolusi, *pooling*, aktivasi dan seterusnya), kemudian digunakan teknik *batch normalization* (*batch norm*) sebagai teknik dalam *neural network* untuk menormalisasikan data *input* dalam ukuran *mini-batch* ketika *training* dilakukan, setelahnya, dilanjutkan dengan penggunaan fungsi aktivasi, dan penambahan *layer* lain hingga mencapai *layer* terakhir, yaitu *layer output*. Tahapan yang ditempuh dalam pembuatan model adalah sebagai berikut [117]:

- 1) Mendefinisikan *layer* yang hendak digunakan dalam model. Setiap *layer* dapat mencakup satu atau lebih konvolusi, *pooling*, batch normalisasi maupun fungsi aktivasi.
- 2) *Add layer*. *Layer* dapat ditambahkan dengan fungsi *add()*.
- 3) *Compile Model*. Mengaplikasikan *loss function* dan *optimizer* sebelum menggunakan fungsi *compile()*
- 4) Melakukan *fitting* model dengan data *train*, *function* yang dapat digunakan adalah *function fit()*.
- 5) Membuat prediksi. Model dapat melakukan prediksi pada data baru dengan menggunakan *function* seperti *evaluate()* dan *predict()*.

2.2.4 Confusion Matrix dan Akurasi

Confusion Matrix merupakan sebuah tolak ukur yang paling umum digunakan untuk mengukur performa model dalam melakukan prediksi. Hasil *confusion matrix* berupa nilai positif dan negatif. Adapun tabel 2.3 merupakan tabel berisi parameter yang digunakan dalam *confusion matrix* [118].

Tabel 2.3 Confusion Matrix

		Predicted	
		Negative	Positive
Actual	Negative	TN	FP
	Positive	FN	TP

Pada tabel 2.3 merupakan *confusion matrix* yang terdiri dari nilai-nilai klasifikasi yang terdiri dari nilai benar (*actual*) dan hasil prediksi (*predicted*) yang dapat digambarkan dalam bentuk nilai sebagai berikut [119] [120]:

- a) *True Positive (TP)*: merupakan jumlah prediksi benar dan menyatakan kelas tersebut memiliki nilai positif. Contohnya, prediksi positif yang benar. Pada rumus 1.7 merupakan rumus persamaan *true positive* atau *sensitivity* atau *recall* yang berarti nilai aktual positif akan menghasilkan nilai positif.

Rumus 1.7 Persamaan True Positive Confusion Matrix

$$\text{True Positive (Sensitivity or Recall)} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

- b) *True Negative (TN)*: merupakan jumlah prediksi benar dan menyatakan kelas tersebut memiliki nilai *negative*. Contohnya, prediksi *negative* yang benar. Pada rumus 1.8 merupakan rumus persamaan *true negative rate (specificity)* dimana terdapat sebuah probabilitas bahwa sebuah nilai aktual *negative* menghasilkan nilai *negative*.

Rumus 1.8 Persamaan True Negative Confusion Matrix

$$\text{True Negative Rate (Specificity)} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}$$

- c) *False Negative (FN)*: merupakan jumlah prediksi salah dan menyatakan kelas tersebut memiliki nilai positif. Contohnya, prediksi *negative* yang salah. Pada rumus 1.9 merupakan rumus persamaan *false negative rate*

dimana terdapat probabilitas saat sebuah nilai *negative* diberikan padahal nilai aslinya adalah *positive*.

Rumus 1.9 Persamaan False Negative Confusion Matrix

$$\text{False Negative Rate} = \frac{\text{False Negative}}{\text{False Negative} + \text{True Positive}}$$

- d) *False Positive (FP)*: merupakan jumlah prediksi salah dan menyatakan kelas tersebut memiliki nilai negatif. Contohnya, prediksi positif yang salah. Pada rumus 1.10 merupakan rumus persamaan *false positive rate* dimana terdapat probabilitas saat sebuah nilai *positive* akan diberikan padahal nilai aslinya adalah *negative*.

Rumus 1.10 Persamaan False Positive Confusion Matrix

$$\text{False Positive Rate} = \frac{\text{False Positive}}{\text{False Positive} + \text{True Negative}}$$

Berdasarkan hasil dari *confusion matrix* kemudian dapat digunakan untuk menghitung nilai *precision*, *recall*, *f1-score* dan *accuracy* (*binary classification*) yang dapat diartikan dan dirumuskan sebagai berikut [121] [122]:

- a) *Precision*

Precision mencari nilai positif dari hasil prediksi bernilai positif. *Precision* menghitung rasio TP dengan jumlah TP dan FP. *Precision* juga merupakan metrik pengukuran yang penting dalam klasifikasi data.

Rumus 1.11 Persamaan Precision

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \times 100\%$$

- b) *Recall (True positive rate)*

Recall menemukan nilai positif yang berhasil diprediksi dari keseluruhan total nilai positif *actual*. *Recall* menghitung TP dengan jumlah TP dan TN. *Recall* penting dalam mengidentifikasi jumlah klasifikasi data yang benar dari keseluruhan data yang diprediksi secara benar.

Rumus 1.12 Persamaan Recall

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \times 100\%$$

Setelah mengetahui nilai *precision* dan nilai *recall*, yang dihasilkan dari *confusion matrix*, kita dapat melihat akurasi performa model dalam

mengidentifikasi objek dan melakukan klasifikasi biner (organik maupun non-organik) pada gambar sampah, dengan menggunakan kedua nilai tersebut untuk dimasukkan ke dalam perhitungan *f1-score*.

c) *F1-score*

F1-score merupakan *harmonic mean* (tingkat rata-rata dalam bentuk pecahan ataupun desimal), dari nilai *precision* dan *recall*. *F1-score* ini mencapai nilai terbaiknya pada poin 1 dan terburuk pada poin 0.

Rumus 1.13 Persamaan *F1-score*

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

F-1 score lebih bermanfaat dibandingkan nilai akurasi [123], namun perhitungan akurasi juga dapat dilakukan setelah mengetahui nilai *confusion matrix*, dapat dihitung melalui rumus 1.11.

d) *Accuracy*, ketika diberikan poin data dengan nilai (TP+TN), akurasi merupakan jumlah rasio prediksi yang benar (TP+TN+FP+FN) pada poin data tersebut. Akurasi sangat penting sebagai pengukur untuk mengevaluasi performa klasifikasi model. Akurasi memiliki nilai ukur terbaik pada poin 1.0 dan terburuk pada poin 0.0.

Rumus 1.14 Persamaan *Accuracy*

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

Keterangan:

TP = Nilai *True Positive*

TN = Nilai *True Negative*

FP = Nilai *False Positive*

FN = Nilai *False Negative*

2.2.4 *Framework Bootstrap*

Bootstrap adalah sebuah *framework* CSS, HTML, dan *JavaScript* untuk membangun sebuah *website* yang modern, responsif dan juga dapat dioperasikan melalui *mobile*. Berikut merupakan beberapa fungsi *bootstrap* [124]:

- a. Mengembangkan website *mobile-friendly*. Situs web yang dibangun dengan komponen UI siap pakai dari *bootstrap* dapat dijalankan secara sempurna di perangkat *mobile*.
- b. Website lebih interaktif dan modern. Adanya *custom plugin* membuat pengguna bebas berkreasi dengan *pop-up* dan elemen interaktif lain yang mendukung pembuatan tampilan agar tampak lebih minimalis dan kekinian.
- c. Menghemat waktu. Dengan menggunakan *framework bootstrap*, tidak perlu membuang waktu mengembangkan *website* dari nol, dan memiliki koleksi elemen bawaan yang mempermudah pekerjaan.

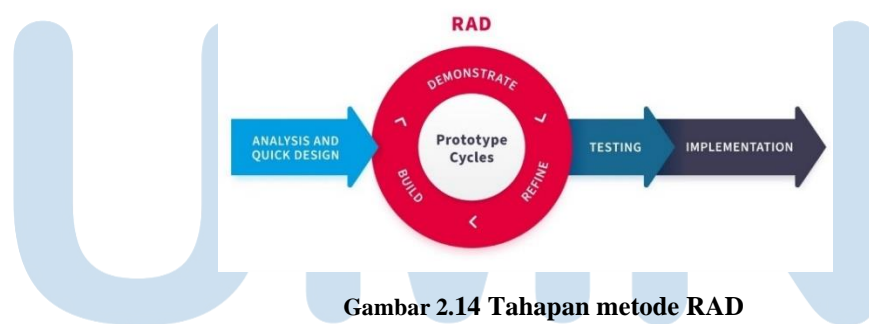
2.2.5 Metode RAD (*Rapid Application Development*)

2.2.5.1 Pengertian RAD

RAD (*Rapid Application Development*) merupakan salah satu metode pengembangan sistem dengan siklus pengembangan yang pendek karena hanya membutuhkan waktu yang singkat. Metode RAD melibatkan *user* dalam penyusunan rencana kebutuhan sistem [125].

2.2.5.2 Tahapan RAD

Secara garis besar, metode RAD dibagi menjadi 3 proses utama, yaitu [125]:



Gambar 2.14 Tahapan metode RAD
Sumber: [126]

1. Requirement Planning

Tahapan ini merupakan tahapan pertama yang ditempuh dalam metode RAD, yaitu identifikasi tujuan dan syarat sistem dilakukan sesuai dengan kebutuhan pengguna. Pengguna dilibatkan

dengan proses ini agar dapat berperan sebagai standar untuk penilaian keberhasilan sistem.

2. *Workshop Desain RAD*

Tahapan ini merupakan tahapan desain perancangan sistem dengan cara menganalisis solusi dari permasalahan pada tahapan sebelumnya. Tahapan ini mencakup tahap perancangan basis data yang akan digunakan dalam sistem. Pada tahapan ini, *user* akan dilibatkan untuk menilai apakah sistem sudah layak untuk digunakan atau belum. Jika belum, perlu kembali dilakukan pemeriksaan dan perbaikan pada sistem.

3. *Implementation*

Tahapan implementasi merupakan tahap finalisasi sistem yang dilanjutkan dengan pengujian pada sistem oleh *user* untuk memeriksa apakah sistem telah berjalan dengan baik dan lancar, apakah sudah sesuai kebutuhan atau belum. Apabila sudah sesuai, sistem dapat diimplementasikan. Jika belum sesuai, maka perlu dilakukan perbaikan pada sistem.

2.2.5.3 Kelebihan RAD

Sebagai sebuah metode pengembangan sistem, RAD memiliki beberapa kelebihan dan kekurangan, antara lain sebagai berikut [127]:

- a. Metode RAD sangat efektif diterapkan karena pembuatan sistem dilakukan dari sisi *user* agar sistem yang dirancang menyesuaikan dengan kebutuhan *user* secara langsung.
- b. Cocok digunakan sebagai metode pengembangan sistem yang memerlukan waktu yang singkat.
- c. Efektivitas waktu lebih singkat dan efisien, karena komponen yang ada dapat digunakan kembali agar *developer* tidak perlu membuat dari awal.

2.2.5.4 Kekurangan RAD

Selain kelebihan, RAD juga memiliki kekurangan, antara lain sebagai berikut [128]:

- a. Komitmen dalam menuntut pengembangan aktivitas untuk melengkapi sistem dalam kerangka waktu yang pendek
- b. Apabila terjadi perubahan dalam pengerjaan sistem, kontrak baru antara pelanggan dengan pengembang harus kembali dibuat, membutuhkan tenaga kerja lebih dalam menangani proyek skala besar
- c. Tidak cocok digunakan untuk sistem dengan resiko teknik yang tinggi karena dapat menjadi hambatan pada sistem yang dirancang.


2.2.6 Unified Modelling Language (UML)

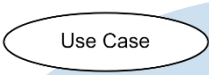

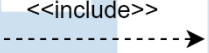
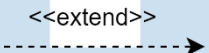

Unified Modelling Language (UML) merupakan sebuah teknik pemodelan diagram yang bertujuan untuk menjelaskan rancangan sistem yang sedang dikembangkan. UML memberikan notasi untuk desain yang berorientasi pada objek. UML bermanfaat dalam membantu pihak *user* dan pihak *developer* sistem dalam memahami secara keseluruhan mengenai sistem yang telah dibuat. 3 diagram penting dalam UML antara lain sebagai berikut [129]:

2.2.6.1 Use Case Diagram

Use Case Diagram adalah diagram yang bertujuan untuk menggambarkan dan menjelaskan struktur dari sebuah sistem, yaitu hubungan antar kelas aktor dan dan atribut kegiatan yang dilakukan. Adapun atribut simbol dalam *use case diagram* beserta fungsinya dijabarkan pada tabel 2.4 [130].

Tabel 2.4 Simbol Use Case Diagram

Simbol	Nama	Keterangan
 Actor	Actor	Pengguna sistem dari eksternal sebagai pihak yang mendapatkan informasi dari sistem tersebut.

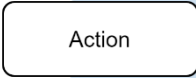
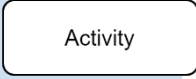


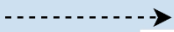
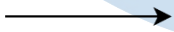

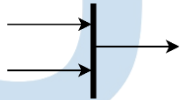
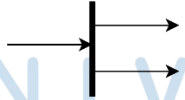

Simbol	Nama	Keterangan
	<i>Use Case</i>	Gambar utama dari kegiatan yang dilakukan oleh <i>actor</i> terhadap sistem yang saling berkaitan dengan <i>use case</i> lainnya.
	<i>Association Relationship</i>	Hubungan actor dengan use case sesuai dengan aktivitasnya masing-masing.
	<i>Include Relationship</i>	Hubungan fungsional antara satu <i>use case</i> dengan <i>use case</i> lain.
	<i>Extend Relationship</i>	Hubungan fungsional dengan adanya perluasan <i>use case</i> yang mencakup kegiatan opsional.
	<i>Generalization Relationship</i>	Mengubah sifat khusus dari sebuah hubungan yang khusus menjadi hubungan yang lebih umum.

Pada tabel 2.4, berisi simbol dan atribut yang digunakan dalam pembuatan *use case diagram* untuk model sistem.

2.2.6.2 Activity Diagram

Activity diagram adalah diagram yang berfungsi untuk menggambarkan alur aktivitas yang dilakukan masing-masing peran terhadap sistem yang telah dibuat. *Activity diagram* dapat digunakan untuk memvisualisasikan *use case* yang lebih spesifik dari *use case* biasa. Adapun atribut dan simbol yang terdapat pada *activity diagram* dijabarkan pada tabel 2.5 [130].

Tabel 2.5 Simbol Activity Diagram

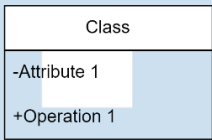
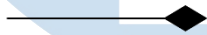



Simbol	Nama	Keterangan
	<i>Action</i>	Aksi yang dilakukan dalam sistem yang tidak dapat diubah lagi.
	<i>Activity</i>	Aktivitas yang dilakukan dalam sistem, biasanya diawali dengan kata kerja.
	Status awal	Status awal dari serangkaian aksi atau <i>activity</i> .
	Status akhir	Status akhir yang mengakhiri serangkaian aksi, objek atau <i>activity</i> .
	<i>Object Flow</i>	Gambaran alur dari <i>action</i> ataupun aksi menuju alur lainnya.
	<i>Control Flow</i>	Gambaran alur urutan pelaksanaan kegiatan dari awal hingga akhir.
	<i>Swimlane</i>	Gambaran pemisahan dikarenakan adanya pengelompokan aktivitas, untuk memecahkan dua actor atau objek.
	<i>Join (Merge)</i>	Menggabungkan aksi atau <i>activity</i> yang bersifat paralel.
	<i>Fork</i>	Memecah aktivitas atau aksi yang paralel menjadi tunggal
	<i>Decision</i>	Gambaran kondisi tertentu bahwa sebuah harus dilakukan pengambilan suatu keputusan atau tindakan.

Pada tabel 2.5, berisi simbol dan atribut yang digunakan dalam pemodelan *activity diagram*.

2.2.6.3 Class Diagram

Class diagram adalah struktur pemodelan sistem yang berisi kelas-kelas yang digunakan. *Class diagram* merepresentasikan bentuk grafis seperti kategori, perilaku, dan hubungan antar kategori. Adapun simbol dan atribut yang digunakan dalam *class diagram* dijabarkan pada tabel 2.6 [130].

Tabel 2.6 Simbol Class Diagram

Simbol	Nama	Keterangan
	<i>Class</i>	Gambaran <i>actor</i> , tempat, ataupun objek yang terlibat dalam sistem.
	<i>Composition</i>	Hubungan fisik yang kuat dari beberapa kelas termasuk kelas itu sendiri.
	<i>Aggregation</i>	Hubungan logis yang kuat dari beberapa kelas
	<i>Association</i>	Hubungan antar kelas
	<i>Generalization</i>	Hubungan umum dari beberapa kelas.

Tabel 2.6 merupakan simbol dan atribut yang digunakan dalam pemodelan struktur *class diagram*.

2.2.7 Black Box Testing

Black box testing merupakan sebuah metode pengujian sistem yang dibuat untuk melihat apakah sistem sudah berjalan sesuai dengan harapan *user*. *Black Box Testing* dilangsungkan untuk mencari kesalahan ataupun *error* pada

sistem yang sekiranya sempat terlewat serta mencari kelemahan sistem yang telah dibuat agar *error* ataupun kesalahan tersebut dapat diatasi sebelum digunakan oleh *user* [131].

2.2.8 User Acceptance Test (UAT)

User Acceptance Test (UAT) adalah sebuah metode pengujian terhadap *user* untuk memeriksa apakah sistem dapat diterima dan berjalan sesuai dengan kebutuhan *user*. Proses UAT dilakukan oleh *user* dengan memeriksa skenario jalannya sistem untuk menguji apakah sistem sudah bebas dari *error* dan layak untuk digunakan. Pengujian ini juga mencakup pengujian terhadap tingkat kepuasan pengguna terkait sistem yang telah dibuat [132].

Dalam menjalankan pengujian UAT, digunakan sebuah perhitungan untuk menghitung nilai persentase keberhasilan fungsi dalam sistem, seperti pada persamaan (7) [133]:

Rumus 1.15 Persamaan Persentase Keberhasilan UAT

$$\text{Persentase keberhasilan} = \frac{\text{Jumlah skenario berhasil}}{\text{Jumlah seluruh skenario}} \times 100\% \quad (11)$$

Setelah mendapatkan persentase keberhasilan, keberhasilan dapat dikategorikan menjadi beberapa kriteria, antara seperti pada tabel 2.7 [133].

Tabel 2.7 Kriteria Interpretasi Skor Keberhasilan UAT

Range Persentase	Keterangan
0% - 20%	Sangat kurang baik
21% - 40%	Kurang baik
41 - 60%	Cukup baik
61% - 80%	Baik
81% - 100%	Sangat baik

2.2.9 Skala Likert

Skala *likert* digunakan sebagai skala pengukuran untuk mengukur persepsi, sikap, ataupun pendapat dari responden terkait sebuah peristiwa ataupun fenomena sosial [134]. Terdapat dua bentuk pertanyaan dalam skala *likert*, yaitu pertanyaan positif (5, 4, 3, 2, 1), dan pertanyaan negatif (skala skor

1, 2, 3, 4, 5). Pengukuran dilakukan dengan memberikan pertanyaan kepada responden yang diminta untuk memilih lima pilihan jawaban. Jawaban yang dapat dipilih adalah sangat setuju (SS), setuju (S), netral (N), tidak setuju (TS), dan sangat tidak setuju (STS) pada pertanyaan positif maupun pertanyaan negatif [135].

Hasil yang diperoleh dari pengukuran skala *likert* kemudian dapat diterjemahkan dalam bentuk kuantitatif agar dapat dihitung, dengan cara pemberian bobot atau skor pada jawaban dari responden. Bobot dari pertanyaan yaitu dimulai dari yang paling kecil, misalnya sangat setuju (SS) = 5, setuju (S) = 4, netral (N) = 3, tidak setuju (TS) = 2, dan sangat tidak setuju (STS) = 1. Untuk menghitung total skor, bobot kemudian dikalikan dengan jumlah responden yang memilih bobot tersebut. Contohnya, terdapat 4 responden yang memilih setuju, maka skornya adalah $4 \times 4 = 16$. Adapun persamaan yang digunakan dalam perhitungan indeks skala *likert*, antara lain sebagai berikut [135].

Rumus 1.16 Persamaan Skala Likert

$$\begin{aligned} \text{Total skor} &= (\text{jumlah responden pilih bobot 1} \times \text{bobot pertanyaan 1}) \\ &+ (\text{jumlah responden pertanyaan 2} \times \text{bobot pertanyaan 2}) \\ &+ \dots \text{dst} \end{aligned}$$

$$\text{Skor maksimum} = (\text{jumlah responden} \times \text{skor tertinggi likert})$$

$$\text{Skor minimum} = (\text{jumlah responden} \times \text{skor terendah likert})$$

$$\text{Indeks (\%)} = \frac{\text{total skor}}{\text{skor maksimum}} \times 100 \quad (12)$$

Dari Indeks skor yang telah didapat, dapat ditentukan dengan penilaian seperti pada tabel 2.8 [135].

Tabel 2.8 Interval penilaian indeks skala *likert*

Indeks	Keterangan
0% - 19,99%	Sangat Tidak Setuju
20% - 39,99%	Tidak Setuju
40% - 59,99%	Netral
60% - 79,99%	Setuju
80% - 100%	Sangat Setuju

2.3 Tools dan Software yang digunakan

2.3.1 Python

Python merupakan sebuah aplikasi yang diciptakan oleh *Guido van Rossum*, di *Netherland*. *Python* pertama kali dirilis dengan versi pertamanya (versi 0.9.0) pada tahun 1991 [136]. Pada tahun-tahun berikutnya, *python* dipilih sebagai bahasa pemrograman karena implementasinya mudah dan sederhana, berbeda dengan teknik pemrograman C dan C++ yang harus di kompilasi terlebih dahulu. *Python* telah terintegrasi seperti halnya ketika ingin memasukkan kode untuk teknik pemrograman yang lain. Kelebihan *python* salah satunya yaitu kode dalam *python* dapat diubah dan dieksekusi secara cepat, dan lebih mudah untuk dipelajari, jika dibandingkan dengan teknik pemrograman lainnya seperti C ataupun *Java* [137].

Beberapa keunggulan dari *software python* yaitu memiliki banyak *framework* dan *library* yang siap dipakai, *multi-platform* dan memiliki komunitas yang besar sehingga peneliti dapat bertanya dan belajar dari peneliti lain pada ruang komunitas [138].

2.3.2 Visual Studio Code

Visual Studio Code (VSCode) merupakan sebuah teks *editor* ringan bersifat *open source* dan handal buatan *Microsoft* yang mendukung operasi *development* aplikasi, seperti *debugging*, *task running*, dan kontrol versi. *VS Code* mendukung bahasa pemrograman *JavaScript*, *Typescript*, *Python*, *C++*, *C#*, dan lain-lain. Pembaruan versi *Visual Studio code* diadakan berkala setiap bulan, oleh karena itu *VSCode* menjadi lebih unggul dan *ter-update* dibandingkan dengan teks editor lainnya [23].

2.3.3 XAMPP

XAMPP merupakan sebuah program *web* lengkap yang merupakan campuran dari beberapa program, seperti basis data *MySQL*, *PHP*, *Apache*, dan server *HTTP*. *XAMPP* memiliki *server* yang dapat berdiri sendiri, yaitu *localhost*. *PHPmyadmin* merupakan bagian dari *XAMPP* untuk melakukan

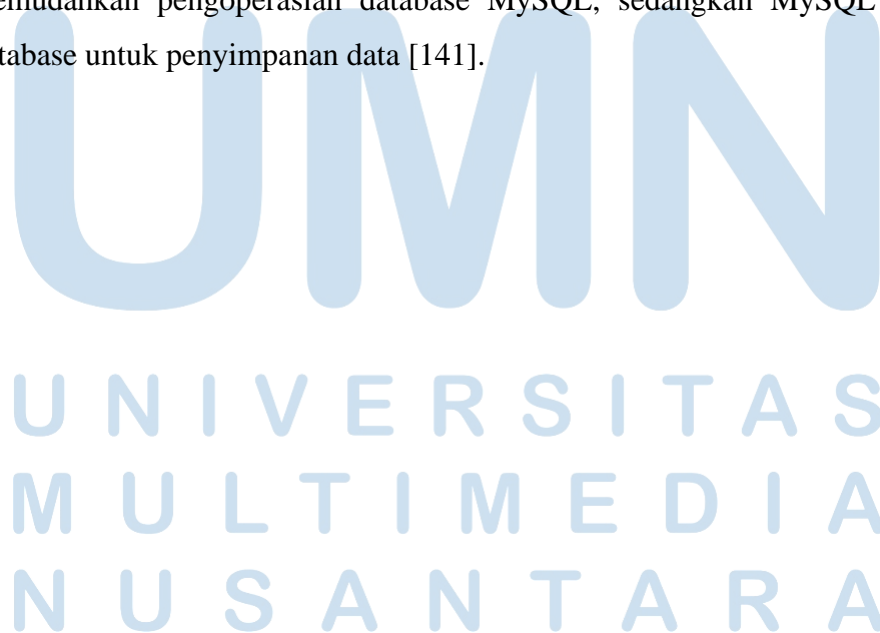
pengolahan data berbasis MySQL untuk menyimpan data secara local. PHPmyadmin dapat diakses melalui *browser*. Aplikasi XAMPP bersifat *open-source* sehingga bebas untuk digunakan [139].

2.3.4 MySQL

MySQL merupakan sebuah server *database* untuk melakukan sistem manajemen basis data relasional (RDBMS) yang juga bersifat *open-source* dan populer. MySQL menggunakan bahasa SQL dalam mengoperasikan *database* yang tersedia melalui platform *windows* dan *linux*. Penggunaan MySQL yaitu dengan menggunakan baris perintah (*command-line interface*), ataupun melalui antarmuka *web* seperti PHPMyAdmin yang dapat memudahkan akses manajemen *database* MySQL [140].

2.3.5 PHPMyAdmin

PHPMyAdmin berperan sebagai *interface* dalam melakukan akses manajemen MySQL. PHPMyAdmin merupakan sebuah aplikasi berbasis *open-source* yang dapat digunakan secara gratis untuk melakukan administrasi data pada database MySQL. PHPMyAdmin mendukung operasi dalam MySQL seperti mengelola basis data, tabel, relasi, bidang (*fields*), pengguna (*users*). PHPMyAdmin berbeda dengan MySQL, PHPMyAdmin adalah alat yang memudahkan pengoperasian database MySQL, sedangkan MySQL adalah database untuk penyimpanan data [141].



2.4 Penelitian Terdahulu

Adapun tabel 2.9 merupakan tabel tentang penelitian yang telah dilakukan sebelumnya oleh penelitian lain yang digunakan sebagai referensi dalam melangsungkan penelitian ini.

Tabel 2.9 Penelitian Terdahulu

No.	Judul Artikel dan Nama Jurnal	Nama Penulis	Metode dan Hasil	Kesimpulan
1.	<i>Trash Image Classification using Machine Learning and Deep Learning Algorithms, Doctoral Dissertation, Dublin, National College of Ireland, 2020.</i>	<p>Penulis: Himanshu Gupta</p> <p>Country: Irlandia</p> <p>Tools: Python</p> <p>Future Research: Menggunakan algoritma untuk mendeteksi objek dan segmentasi untuk mencapai nilai akurasi yang lebih tinggi dengan dataset yang lebih setara dan ukuran data yang lebih besar untuk meningkatkan skalabilitas, memperluas kategori sampah agar tidak hanya terbatas pada 5 kategori, namun lebih banyak dengan gambar <i>multilabel</i>.</p>	<p>Metode: Pengumpulan dan <i>pre-processing</i> data (sebanyak 3,442 data sampah), analisa data dan augmentasi, membuat model <i>sequential</i> keras, ResNet-50, VGG-19, dan model XGBoost sebagai algoritma dalam <i>machine learning</i> dan <i>deep learning</i> untuk melakukan klasifikasi dan prediksi gambar sampah pada <i>dataset</i> TACO.</p> <p>Model yang digunakan adalah Model <i>Sequential Keras</i> (dengan API Keras, <i>batch size</i> 50, pembagian <i>validation</i> 20% yaitu 551 gambar <i>validation</i> dan 2,202 <i>data training</i>, serta <i>early stopping patience</i> 20), ResNet-50 (3 <i>layer</i> yaitu <i>pool</i>, <i>convolutional</i>, dan <i>average pool layer</i>), VGG-19 (16 <i>convo layer</i>, 3 <i>fully connected layer</i>, 5 <i>max pooling layer</i>, dan 1 <i>softmax layer</i>), dan XGBoost (50 <i>boost round</i> dan 10 <i>early stopping</i>).</p> <p>Hasil:</p>	Mengimplementasikan beberapa model <i>Deep Learning</i> untuk melakukan klasifikasi prediksi pada gambar sampah dalam dataset TACO. Ditemukan bahwa model <i>pre-trained</i> memiliki performa yang lebih baik dibandingkan model yang dibangun dari awal seperti model <i>sequential Keras</i> karena cenderung sederhana sehingga performa model tidak sebaik <i>pre-trained models</i> .

No.	Judul Artikel dan Nama Jurnal	Nama Penulis	Metode dan Hasil	Kesimpulan
			<p>Dengan menggunakan model <i>Sequential</i> Keras diperoleh akurasi sebesar 83-89.9% namun model kurang berhasil dan memprediksi data baru.</p> <p>Model <i>pre-trained ResNet-50</i> memiliki akurasi sebesar 85.9% pada dataset yang besar, sedangkan model VGG-19 memiliki akurasi test sebesar 86.4%, dan <i>XGBoost</i> memiliki akurasi 69%.</p> <p>Dapat disimpulkan bahwa model <i>pre-trained</i> seperti <i>ResNet-50</i> dan VGG19 memiliki performa yang lebih baik dibandingkan model lainnya.</p>	
2.	<i>ScrapNet: An Efficient Approach to Trash Classification, IEEE Access, vol. 9, 2021.</i>	<p>Penulis: Masand Abhishek, Suryansh Chauhan, Mahesh Jangid, Rajesh Kumar, Satyabrata</p> <p>Country: India Tools: Python Future Research: -</p>	<p>Metode: Melakukan eksperimen pada berbagai model untuk dikomparasi performa dan akurasinya. Dataset yang digunakan merupakan data gabungan antara data TACO (1,501), <i>OpenRecycle</i> (2,301), <i>TrashNet</i> (2,527), dan <i>waste classification</i> (850). Dataset gabungan ini kemudian dijadikan sebagai data untuk melatih model baru yang akan dibuat yaitu model "<i>Scrap Net</i>".</p> <p>Pembagian data yang digunakan adalah 80 <i>training</i>, 10 <i>validation</i>, dan 10 <i>testing</i>. Dataset juga diujikan pada model lain seperti <i>Efficient Net B0</i> dan <i>B3</i> dengan <i>optimizer</i> RMSProp, <i>learning rate</i> 0.0001.</p>	Penelitian dataset TACO dengan menggunakan model baru yang dibangun oleh peneliti pada jurnal ini dengan nama " <i>Scrap Net</i> ", dengan tingkat akurasi mencapai 98%.

No.	Judul Artikel dan Nama Jurnal	Nama Penulis	Metode dan Hasil	Kesimpulan
			<p>Hasil: Ketika data diuji, model lain, model <i>Efficient Net</i> B0 memiliki akurasi terbaik pada angka 92.08%, <i>Efficient Net</i> B3 dengan akurasi 92.87%, ResNet50 83.11%, dan model baru yang dibuat dalam penelitian ini (<i>Scrap Net</i>) mendapatkan akurasi sebesar 98.4%. Akurasi <i>Scrap Net</i> unggul dibandingkan dengan akurasi model lain seperti SVM (63%), <i>ResNet</i> (88%), dan <i>DenseNet121</i> (95%).</p>	
3.	<p><i>Aspect-Based Sentiment Analysis on Application Review using Convolutional Neural Network</i>, Ultima InfoSys: Jurnal Sistem Informasi, vol. 13, No. 1, 2022. (UMN Affiliation)</p>	<p>Penulis: Putri Arta Aritonang, Monika Evelin Johan, Iwan Prasetiawan</p> <p>Country: Indonesia Tools: - Future Research:</p>	<p>Metode: Mengimplementasikan algoritma CNN sebagai model untuk melakukan klasifikasi aspek dan sentimen. Data yang digunakan adalah data <i>review</i> dari <i>Google Play store</i>. Data dibagi menjadi 70% data <i>train</i>, 15% data <i>validation</i>, dan 15% data <i>test</i>.</p> <p>Hasil: Berdasarkan hasil evaluasi model CNN pada Model klasifikasi aspek, didapatkan hasil akurasi <i>training</i> sebesar 98.5% dan akurasi validasi 90.2%. Untuk klasifikasi teks <i>sentiment</i> didapatkan akurasi sebesar 97.4% dan akurasi validasi 93.7%. Pengukuran yang digunakan dalam penelitian ini adalah <i>f1-score</i>. <i>F1-score</i> yang didapat dari klasifikasi aspek adalah 92.23%, dan 95.12% dalam klasifikasi sentimen.</p>	<p>Penerapan CNN sebagai algoritma dengan performa yang baik selain di bidang analisis gambar, penelitian ini membuktikan dalam bidang analisis sentimen.</p>

No.	Judul Artikel dan Nama Jurnal	Nama Penulis	Metode dan Hasil	Kesimpulan
4.	<i>Waste Image Classification Based on Transfer Learning and Convolutional Neural Network, Waste Management, vol. 135, pp.150-157, 2021.</i>	<p>Penulis: Zhang Qiang, Qifan Yang, Xujuan Zhang, Qiang Bao, Jinqi Su, Xueyan Liu</p> <p>Country: China</p> <p>Tools: Python</p> <p>Future Research: Mengkombinasikan metode transfer learning untuk mengimprovisasi CNN sehingga model dapat berkembang menjadi lebih efektif dalam mengidentifikasi jenis sampah, dan meningkatkan akurasi dan efisiensi dari model.</p>	<p>Metode: Mengimplementasikan model CNN dengan metode <i>transfer learning</i> pada dataset TrashNet berjumlah 18,911 gambar non-organik seperti plastik, karton, besi, gelas, dan kertas. Pembagian data dengan skala 70:30 untuk training dan testing. Model yang digunakan untuk klasifikasi adalah model DenseNet169, dan <i>activation function</i> ReLU. Model kemudian dievaluasi dengan menggunakan <i>confusion matrix</i>.</p> <p>Hasil: Setelah melakukan training model dengan 300 <i>epochs</i> pada penggabungan DenseNet169 dengan transfer learning, akurasi yang didapat yaitu sebesar 82.80%. Hal ini membuktikan bahwa penggabungan kedua metode (DenseNet169 + <i>Transfer Learning</i>) menghasilkan akurasi yang lebih tinggi dibandingkan hanya menggunakan satu metode yaitu DenseNet169 yang hanya menghasilkan akurasi sebesar 47.83%.</p>	Mengimplementasikan model CNN dengan arsitektur DenseNet169 dan <i>transfer learning</i> dengan 300 <i>epochs</i> . Akurasi yang didapat relatif tinggi sehingga membuktikan bahwa CNN merupakan algoritma yang tepat untuk melakukan klasifikasi gambar.
5.	<i>A Study of Garbage Classification with Convolutional Neural Networks,</i>	<p>Penulis: Meng Shanshan, Wei Ta Chu</p> <p>Country: Taiwan</p>	<p>Metode: Menggunakan <i>simple CNN architecture</i> dengan 2D <i>convolutional layers</i> dengan ReLU sebagai <i>activation function</i>. Model CNN lain seperti ResNet50, HOG (<i>Histogram of Oriented Gradients</i>) +CNN, SVM, dan <i>Plain network</i> ResNet50, juga</p>	Implementasi teknik <i>deep learning</i> untuk menyelesaikan permasalahan klasifikasi gambar

No.	Judul Artikel dan Nama Jurnal	Nama Penulis	Metode dan Hasil	Kesimpulan
	<i>In 2020 Indo-Taiwan 2nd International Conference on Computing, Analytics and Networks (ICAN), 2020.</i>	Tools: - Future Research: Karena banyaknya variasi kategori sampah di kehidupan nyata, model memerlukan sumber data yang lebih besar dan terklasifikasi secara benar, pada situasi yang berbeda.	<p>dibuat untuk dijadikan sebagai perbandingan. Optimizer yang digunakan ada dua, yaitu Adadelata (RMSProp + <i>Momentum</i>), dan Adam.</p> <p>Hasil: Berdasarkan hasil pengujian CNN pada keempat model yang berbeda, diperoleh hasil akurasi test dengan SVM sebesar 47.25%, ResNet50 menghasilkan 95% (Adadelata) dan 51% (Adam). Sementara model <i>plain network</i> untuk ResNet50 76% (Adadelata), dan 82% (Adam). Sementara simple CNN menghasilkan 82% akurasi test (Adam), dan 95% akurasi test (Adadelata). Meskipun <i>optimizer</i> Adam merupakan <i>optimizer</i> yang lebih maju dibandingkan Adadelata, Namun training model yang dihasilkan tidak lebih baik. Namun, dengan menambah data dan meningkatkan <i>epoch</i> diyakini dapat meningkatkan akurasi model dalam melakukan prediksi maupun klasifikasi.</p>	sampah dengan akurasi yang tinggi dengan menggunakan CNN sebagai pendekatan dalam melakukan klasifikasi gambar.
6.	<i>Application of Convolutional Neural Network based on transfer learning for garbage classification, In</i>	Penulis: Cao Li, Wei Xiang Country: China Tools: Python Future Research:	Metode: Menggunakan model <i>Inception-V3</i> dengan menggunakan dataset gambar sampah untuk membuat <i>list</i> dari gambar <i>training</i> , membagi menjadi tiga bagian yaitu 8:1:1 (<i>training, validation, dan testing</i>), kemudian menggunakan <i>train bottleneck</i> sebagai <i>input</i> dari <i>fully connected layer</i> yang terbentuk. Data	<i>Transfer learning</i> digunakan untuk melatih model secara spesifik untuk mengidentifikasi kategori sampah berdasarkan model

No.	Judul Artikel dan Nama Jurnal	Nama Penulis	Metode dan Hasil	Kesimpulan
	2020 IEEE 5 th Information Technology and Mechatronics Engineering Conference (ITOEC). Pp 1032-1036, IEEE, 2020.	Menambahkan lebih banyak data sebagai <i>input</i> untuk <i>network</i> agar dapat meningkatkan model klasifikasi sampah. Bisa juga ditambahkan <i>voice broadcast module</i> dan <i>recognition semantic module</i> untuk melengkapi klasifikasi sampah sehingga penelitian tidak hanya berfokus pada perspektif dari gambar.	<i>bottleneck</i> digunakan untuk memverifikasi dan mengetes akurasi. Hasil: Hasil akurasi yang diperoleh yaitu dengan akurasi tertinggi sebesar 93.2% dengan <i>learning rate</i> =0.1, sedangkan <i>learning rate</i> =0.01 hanya memiliki akurasi sebesar 90.4%, 0.001 sebesar 86.4%, dan 0.0005 sebesar 85.1%. Akurasi terbesar diperoleh kategori “ <i>recyclable waste</i> ” dengan akurasi model terhadap data <i>test</i> sebesar 98%. Dengan demikian, dapat disimpulkan bahwa nilai <i>learning rate</i> terbaik dari model yang di <i>training</i> adalah ketika nilai <i>learning rate</i> -nya bernilai 0.1.	<i>Inception-V3</i> CNN. Melalui pembelajaran dan modifikasi model, <i>rate recognition</i> meningkat menjadi lebih baik daripada aslinya, sehingga model <i>garbage</i> klasifikasi dapat tercipta dengan baik.
7.	Pembangunan Sistem Informasi Perpustakaan Berbasis Web menggunakan Metode <i>Rapid Application Development</i> (RAD) (Studi pada: SMK Negeri 11 Malang), Jurnal	Penulis: Aini Nur, Satrio Agung Wicaksono, Issa Arwani Country: Indonesia Tools: - Future Research: Mengembangkan sistem dengan menambahkan fitur yang lebih kompleks serta tampilan <i>interface</i> yang	Metode: Menggunakan metode RAD dalam merancang sistem perpustakaan berbasis web, serta melakukan pengujian dengan metode <i>black-box</i> , UAT (<i>User Acceptance Testing</i>), dan skala <i>likert</i> . Hasil: Dari hasil UAT (<i>User Acceptance Testing</i>), didapatkan 84% pengguna setuju bahwa sistem memenuhi kebutuhan pengguna. Dari hasil	Mengimplementasikan metode RAD untuk merancang sistem berbasis web, sebuah sistem perpustakaan yang memudahkan admin dalam pengolahan data dan memudahkan <i>user</i> dalam melakukan peminjaman buku.

No.	Judul Artikel dan Nama Jurnal	Nama Penulis	Metode dan Hasil	Kesimpulan
	Pengembangan Teknologi Informasi dan Ilmu Komputer, 2019.	lebih menarik untuk meningkatkan pelayanan perpustakaan	implementasi sistem informasi perpustakaan, tidak ada lagi perbaikan karena sistem telah disempurnakan.	
8.	<i>Waste Classification System using Image Processing and Convolutional Neural Networks, Advances in Computational Intelligence: 15th International Work-Conference on Artificial Neural Networks, pp. 350-36). Springer International Publishing, 2019.</i>	<p>Penulis: Bobulski Janusz, Mariusz Kubanek</p> <p>Country: Polandia</p> <p>Tools: Python</p> <p>Future Research: Melakukan penambahan <i>database</i> dari gambar sampah yang telah dipisahkan dan lebih realistis.</p>	<p>Metode: Mengimplementasikan CNN untuk melakukan klasifikasi dan pengenalan gambar sehingga dapat menjadi sistem klasifikasi yang otomatis dengan menggunakan metode <i>artificial intelligence</i> dan <i>image processing</i>. Model CNN yang dibuat ada 2, yaitu dengan 23 layers dan 15 layers. Jumlah data yang digunakan untuk melatih model adalah sebanyak 33,000 gambar sampah plastik, dengan pembagian data dalam <i>splitting</i> sebanyak 90:10, 80:20, 70:30, dan 60:40.</p> <p>Hasil: Akurasi yang didapatkan dari hasil training model CNN dengan 15 <i>layer network</i> pada pembagian 90:10, dengan 10 <i>epochs</i>, dan resolusi gambar 120x120 pixels, dengan nilai akurasi sebesar 99.92%. Untuk 23 <i>layer network</i>, model CNN memperoleh nilai akurasi tertinggi pada angka 99.23%, dengan jumlah <i>epoch</i> yang sama yaitu 10 <i>epochs</i> namun dengan resolusi gambar yang berbeda yaitu 227 x 227 pixels.</p>	Model CNN berhasil mengklasifikasi gambar dengan baik dengan nilai akurasi yang tinggi.

No.	Judul Artikel dan Nama Jurnal	Nama Penulis	Metode dan Hasil	Kesimpulan
9.	<i>Waste Classification using ANN, CNN, and Transfer Learning, CNN and Transfer Learning SSRN, 2022.</i>	<p>Penulis: Patil Manoj, Naumanurrahman Shaikh, Manish Sanjay Zalte</p> <p>Country: India Tools: Python Future Research: -</p>	<p>Metode: Menggunakan algoritma ANN, CNN, dan <i>Transfer Learning</i> untuk dibandingkan dan dicari model dengan performa terbaik. Model CNN yang digunakan adalah model <i>Sequential Keras</i> dengan melakukan <i>preprocessing</i> gambar terlebih dahulu, seperti membagi data menjadi sampah organik dan sampah non-organik, kemudian membuat model CNN dengan beberapa layer, yang terdiri dari layer konvolusi, layer <i>pooling</i>, dan layer <i>dense</i>. Dengan jumlah <i>epoch</i> yang digunakan adalah 20 <i>epoch</i>, 22.564 data <i>training</i> dan 2.513 data <i>testing</i>, serta <i>activation function</i> yang digunakan adalah ReLU.</p> <p>Hasil: ANN bukan pilihan yang baik untuk melakukan klasifikasi gambar karena nilai akurasi yang relatif rendah (71.76%) dengan 2 <i>layer</i>. Namun, CNN merupakan algoritma yang baik untuk klasifikasi gambar dengan tingkat akurasi mencapai 99.21%.</p>	Mengimplementasikan algoritma ANN, dan CNN sederhana dengan menggunakan model <i>sequential Keras</i> dan <i>Library Tensorflow</i> .
10.	<i>Waste Classification for Sustainable Development Using Image Recognition with Deep Learning</i>	<p>Penulis: Malik Meena, Sachin Sharma, Mueen Uddin, Chin-Ling chen, Chih-Ming Wu, Punit Soni, Shikha Chaudhary</p>	<p>Metode: Menggunakan arsitektur CNN dengan <i>pre-trained</i> model <i>EfficientNet-B0</i> yang dibuat oleh peneliti komputer MingXing Tan dan Quoc V. Le. Dengan <i>dataset</i> berjumlah 23.628 <i>file</i>, mencakup <i>dataset training</i> dan <i>testing</i>, untuk mengklasifikasikan</p>	Mengimplementasikan model CNN <i>pre-trained</i> untuk melakukan klasifikasi pada gambar, dan didapatkan

No.	Judul Artikel dan Nama Jurnal	Nama Penulis	Metode dan Hasil	Kesimpulan
	<i>Neural Network Models, Sustainability</i> MDPI Vol.14 No.12, 2022.	<p>Country: India Tools: Python Future Research: Mengembangkan model untuk mengidentifikasi sub-kelas baru, seperti mengembangkan kelas “karton” menjadi kelas besar, medium, kecil. Contohnya kayu, karton, plastik, dan sebagainya. Mengimplementasikan model pada lengan robot dan HD camera dengan alur operasi yang efisien terhadap seluruh area dengan kandungan sampah padat.</p>	<p>sampah menjadi beberapa kategori. Model <i>Efficient-B0</i> menggunakan referensi ImageNet (akurasi 84.3%) dengan 1000 kelas objek untuk klasifikasi sampah padat. Model memiliki 237 <i>layer</i> yang dibagi menjadi lima modul.</p> <p>Hasil: Didapatkan nilai akurasi sebesar 81.2%, dengan dimensi 224x224 pixels dan tiga buah channel warna, yaitu RGB (<i>red, green, blue</i>). Dapat disimpulkan bahwa model memiliki akurasi yang cukup baik dalam mengklasifikasikan sampah padat.</p>	nilai akurasi yang cukup baik.
11.	Perbandingan <i>Convolutional Neural Network</i> pada <i>Transfer Learning Method</i> untuk	<p>Penulis: Daniel Martomanggolo Wonohadidjojo Country: Indonesia Tools: -</p>	<p>Metode: Menggunakan arsitektur CNN dengan model <i>AlexNet, GoogleNet, ResNet-50, dan VGG-16 (Visual Geometry Group)</i>, dengan metode evaluasi <i>confusion matrix</i> dengan <i>performance metrics</i> adalah <i>accuracy</i>,</p>	Mengimplementasikan beragam model CNN untuk mengklasifikasikan gambar sel darah putih.

No.	Judul Artikel dan Nama Jurnal	Nama Penulis	Metode dan Hasil	Kesimpulan
	Mengklasifikasikan Sel Darah Putih, <i>Ultimatics: Jurnal Teknik Informatika</i> pp. 51-57, Vol.13 No.1, 2021 (UMN Affiliation)	Future Research: Meningkatkan akurasi dan nilai <i>performance metrics</i> dengan menggunakan arsitektur <i>network</i> yang lain dengan menerapkan teknik atau algoritma optimasi.	<i>precision, recall</i> , dan <i>F-measure</i> . Metode klasifikasi yang digunakan adalah <i>transfer learning</i> . Hasil: Didapatkan performa terbaik dalam mengklasifikasikan gambar sel darah putih dihasilkan oleh model ResNet-50, dengan nilai 94.17%, disusul oleh <i>GoogleNet</i> (93.33%), VGG-16 (91.67%), dan <i>AlexNet</i> (90%).	

UMMN

UNIVERSITAS

MULTIMEDIA

NUSANTARA

Berdasarkan penelitian terdahulu pada tabel 2.9, artikel jurnal berjudul “*Trash Image Classification Using Machine Learning and Deep Learning Algorithms*” (akurasi tertinggi 83-89.9%) dan “*ScrapNet: An Efficient Approach to Trash Classification*” (akurasi 92.87%), keduanya menggunakan dataset TACO dalam penelitiannya, sehingga kedua jurnal tersebut dijadikan sebagai referensi penggunaan dataset sampah TACO dalam penelitian ini.

Artikel “*Waste Image Classification Based on Transfer Learning and Convolutional Neural Network*” (Akurasi DenseNet169 82.8%), “*Perbandingan Convolutional Neural Network pada Transfer Learning Method untuk Mengklasifikasikan Sel Darah Putih*” (akurasi arsitektur model ResNet-50 sebesar 94.17%) menjadi dasar pemilihan algoritma CNN sebagai algoritma dengan performa yang baik dalam melakukan deteksi objek dan klasifikasi gambar. Artikel “*Aspect-Based Sentiment Analysis on Application Review Using CNN*” (Akurasi train aspect 98.5% dan akurasi train sentimen 93.7%), mendukung bahwa CNN memiliki performa yang baik di bidang lain selain klasifikasi gambar.

Sementara artikel berjudul “*Waste Classification System Using Image Processing and Convolutional Neural Networks*”, digunakan sebagai referensi dalam menggunakan algoritma CNN sebagai algoritma yang dapat memberikan hasil yang optimal (akurasi 99.92%) dalam melakukan *image processing*.

Artikel “*Pembangunan Sistem Informasi Perpustakaan Berbasis Web menggunakan Metode Rapid Application Development (RAD) (Studi pada: SMK Negeri 11 Malang)*” menginspirasi pembuatan sistem informasi dengan metode RAD yang juga diterapkan dalam penelitian ini sehingga dapat dijadikan referensi.

Kemudian artikel “*A Study of Garbage Classification with Convolutional Neural Networks*” (Akurasi CNN 82%), artikel “*Waste Classification for Sustainable Development Using Image Recognition with Deep Learning Neural Network Models*” (Akurasi model Efficient-B0 81.2%) dan “*Application of Convolutional Neural Network Based On Transfer Learning For Garbage Classification*” (akurasi 93.2%), dijadikan sebagai referensi untuk melakukan

penelitian terkait sampah dan jenisnya yaitu sampah organik dan non-organik untuk dibuatkan sistem klasifikasinya dengan mengimplementasi algoritma CNN.

Terakhir, artikel "*Waste Classification Using ANN, CNN, And Transfer Learning*" digunakan sebagai referensi penggunaan model sequential keras dalam algoritma CNN dalam penelitian ini karena performa model dengan akurasi yang baik (99.21%) dan sederhana dibandingkan dengan model kompleks CNN lainnya. CNN dikenal memiliki nilai akurasi lebih baik dibandingkan dengan ANN (71.76%).

Dengan demikian, dikarenakan penelitian sebelumnya menggunakan algoritma CNN dan memperoleh akurasi yang tinggi dan performa yang baik dalam melakukan banyak kasus klasifikasi gambar, maka metode CNN diadopsi dalam penelitian ini dengan menggunakan model keras Sequential (bukan *pre-trained models*). Adapun alasan tidak menggunakan *pre-trained models* adalah karena umumnya *pre-trained models* merupakan hasil karya peneliti lain yang disimpan ketika dilakukan pelatihan terhadap sebuah dataset besar, juga dibutuhkan akses khusus jika ingin mengakses model *pre-trained* tersebut. Belum lagi, bentuk implementasi model *pre-trained* bisa saja berbeda jika dataset yang digunakan berbeda.

Penelitian pada dataset TACO sebelumnya dilakukan oleh Himanshu Gupta (2020), menggunakan model *pre-trained* seperti ResNet-50 dan VGG19 untuk melakukan klasifikasi model dengan tingkat akurasi diatas 85%. Penelitian lain yang dilakukan dengan pembuatan model baru oleh peneliti Masand Abhishek et al. (2020) dengan nama model "*ScrapNet*" juga memiliki tingkat akurasi sebesar 98%. Dikarenakan akurasi pengujian pada data TACO sudah relative tinggi dengan beragam penggunaan *pre-trained models* maupun model yang dibuat sendiri, pada penelitian ini menggabungkan dataset TACO dengan dataset *garbage* data yang belum pernah dibuatkan model *pre-trained* oleh peneliti sebelumnya, untuk dilihat akurasi model *sequential*, sebagai unsur kebaruan dari penelitian. Selain itu, penelitian sebelumnya hanya melakukan prediksi objek pada gambar sampah dalam

dataset TACO berdasarkan bahan sampahnya, yaitu sampah botol, plastik, kaleng, gelas, besi, dan seterusnya. Namun, belum ada penelitian yang dapat mengklasifikasikan apakah sampah tersebut termasuk ke dalam jenis organik maupun non-organik. Oleh karena itu, peneliti melangsungkan penelitian untuk merancang sebuah sistem klasifikasi gambar berbasis *web* yang dapat mengklasifikasikan gambar organik dan non-organik.

