BAB II

TINJAUAN LITERATUR

2.1 Penelitian Terkait

Penelitian terdahulu menunjukkan bahwa arsitektur MobileNetV2 memiliki kemampuan yang baik dalam melakukan klasifikasi citra sampah. Melalui pendekatan *Transfer Learning* CNN, Sean Edbert Thio dan Joko Susilo (2025) membuktikan bahwa MobileNetV2 lebih superior dibandingkan EfficientNetB0 dengan capaian akurasi training 87,31% serta akurasi validation 91,57%. Efisiensi pemrosesan dari arsitektur ini juga ditekankan oleh Fifi Andriani dkk (2026) ketika mengklasifikasikan sampah jenis kaca, kertas, dan plastik, yang menghasilkan tingkat validasi terbaik sebesar 85,35%. Di samping itu, keandalan model ini teruji lewat studi Valensia Fernanda Ham Ayomi (2024) yang berhasil memetakan 14 kategori sampah anorganik dengan akurasi pengujian mencapai 82% sebelum akhirnya diimplementasikan ke aplikasi mobile menggunakan TensorFlow Lite.

Keunggulan MobileNetV2 dipertegas kembali oleh Muhammad Hanif dan Zaenal Syamsul Arief (2025) melalui studi komparasi dengan CNN Sequential untuk mengidentifikasi empat kelompok sampah. Dari pengujian tersebut, MobileNetV2 memimpin dengan tingkat akurasi sebesar 88%, jauh melampaui model sekuesial yang hanya mencapai 71%. Optimalisasi arsitektur ini untuk memisahkan rumpun organik dan anorganik dari citra digital juga diperlihatkan oleh Syarif dkk (2025) dengan perolehan akurasi validasi yang tinggi sebesar

93%. Rangkaian temuan ini mengukuhkan posisi MobileNetV2 sebagai arsitektur yang tangguh dalam ekstraksi fitur visual dan sangat efisien secara komputasi, menjadikannya opsi terbaik untuk integrasi pada perangkat dengan sumber daya terbatas.

Meskipun demikian, penelitian terdahulu masih memiliki beberapa keterbatasan. Sebagian besar penelitian menggunakan dataset publik dan melakukan pengujian dalam kondisi yang terkontrol, sehingga kemampuan model untuk mengenali data pada kondisi yang berbeda masih perlu ditingkatkan. Selain itu, implementasi klasifikasi sampah secara langsung menggunakan kamera dan berjalan secara *real-time* masih belum banyak dilakukan. Beberapa penelitian juga hanya mengklasifikasikan jenis sampah tertentu, seperti sampah anorganik atau sampah daur ulang, sehingga belum mencakup kebutuhan pemilahan sampah yang umum ditemui dalam kehidupan sehari-hari. Oleh karena itu, penelitian ini menerapkan arsitektur MobileNetV2 pada sistem klasifikasi sampah organik dan non-organik berbasis Computer Vision yang dapat melakukan identifikasi secara otomatis dan *real-time* melalui kamera. Dengan demikian, sistem yang dikembangkan diharapkan mampu memberikan hasil klasifikasi yang lebih praktis dan sesuai dengan kondisi penggunaan di lingkungan nyata.

2.2 Sampah

Menurut *World Health Organization* (WHO), sampah merupakan segala sesuatu yang sudah tidak dimanfaatkan, tidak digunakan, tidak diinginkan, atau dibuang sebagai hasil dari aktivitas manusia dan bukan terbentuk secara alami

(Mita Defitri, 2023). Sedangkan menurut Undang-Undang (UU) Nomor 18 Tahun 2008 tentang Pengelolaan Sampah, sampah diartikan sebagai sisa kegiatan sehari-hari manusia dan/atau proses alam yang berbentuk padat serta memerlukan pengelolaan yang sistematis dan berkelanjutan agar tidak menimbulkan dampak negatif terhadap kesehatan masyarakat maupun lingkungan.

Sampah juga didefinisikan sebagai bahan yang dihasilkan dari aktivitas manusia maupun proses alam yang tidak lagi memiliki manfaat. Keberadaan sampah menjadi persoalan lingkungan yang penting untuk diperhatikan karena berkaitan dengan pengelolaan serta dampak yang dapat memengaruhi kehidupan masyarakat (Syarifah et al., 2022). Secara umum, sampah terbagi menjadi dua jenis, yaitu sampah organik (basah) yang mudah terurai secara alami dan sampah an-organik (kering) yang lebih sulit terurai sehingga membutuhkan penanganan khusus agar tidak merusak lingkungan (Apriliana et al., 2022).

Sampah organik merupakan limbah yang berasal dari sisa-sisa makhluk hidup dan memiliki kemampuan untuk terurai secara alami tanpa bantuan manusia. Jenis sampah ini dapat dimanfaatkan kembali menjadi produk yang berguna apabila dikelola dengan baik. Namun, jika penanganannya tidak tepat, sampah organik dapat menimbulkan bau tidak sedap serta berpotensi menjadi sumber penyakit akibat proses pembusukan yang berlangsung cepat (Amir et al., 2021). Sedangkan sampah non-organik merupakan jenis limbah yang tidak dapat terurai oleh bakteri sehingga membutuhkan waktu yang sangat lama untuk terdegradasi. Umumnya, sampah tersebut berasal dari kegiatan rumah tangga maupun industri. Jika tidak

dikelola dengan baik, keberadaannya dapat menimbulkan dampak negatif bagi lingkungan serta membahayakan kesehatan manusia (Hakiki, 2023).

2.3 *Computer Vision*

Computer vision adalah ilmu komputer yang mengeksplorasi kemampuan visual manusia. *Computer vision* memiliki banyak bidang, salah satunya adalah deteksi objek. Ini bekerja dengan mengenali objek yang ada pada gambar dan letaknya, dan dapat digunakan untuk berbagai tujuan, seperti keamanan, informasi, pengawasan, tingkat produktivitas, dan sebagainya (Dompeipen & Sompie, 2020). *Computer Vision* memiliki kemampuan untuk melihat data dalam bentuk pixel dengan berbagai warna, membandingkan dua objek dalam gambar yang sama, menemukan objek yang lebih jelas, dan membantu meringankan pekerjaan manusia. Di sisi lain, Visi adalah sebuah sistem penilaian informasi yang diambil dari gambar yang biasanya diambil oleh kamera, dengan strategi sekstrasi menggunakan algoritma tertentu. *Computer Vision* akan mempercepat proses produksi dan menghasilkan produk yang cepat dan stabil (Bagus et al., 2024).

Tujuan dari *computer vision* tidak hanya sebatas menangkap atau melihat gambar, tetapi juga mengolahnya serta menghasilkan informasi yang bermanfaat berdasarkan hasil analisis tersebut (Rosalina & Wijaya, 2020). Dalam prosesnya, *Computer Vision* memanfaatkan berbagai teknik pengolahan citra dan pembelajaran mesin untuk mengenali karakteristik objek yang terdapat pada citra. Salah satu penerapan *Computer Vision* yang banyak digunakan adalah klasifikasi

citra (*image classification*), yaitu proses mengelompokkan suatu citra ke dalam kategori tertentu berdasarkan karakteristik visual yang dimilikinya.

2.4 *Deep Learning*

Deep Learning merupakan salah satu cabang dari *Machine Learning* yang berada dalam lingkup *Artificial Intelligence* (AI). *Deep Learning* memanfaatkan jaringan saraf tiruan (*Artificial Neural Network*) yang terinspirasi dari cara kerja otak manusia untuk mempelajari pola dan karakteristik dari sejumlah besar data (Rosalina & Wijaya, 2020). Pendekatan ini memungkinkan sistem melakukan proses pembelajaran secara otomatis tanpa memerlukan ekstraksi fitur secara manual sebagaimana pada beberapa metode *Machine Learning konvensional*.

Perkembangan *Deep Learning* telah mendorong kemajuan berbagai aplikasi kecerdasan buatan, seperti pengenalan suara, pengolahan citra, bidang kesehatan, serta *Computer Vision*. Kemampuannya dalam mengekstraksi fitur dan mengenali pola secara otomatis menjadikan *Deep Learning* banyak diterapkan dalam berbagai tugas, termasuk deteksi objek, klasifikasi citra, segmentasi gambar, dan pengenalan wajah (Informatika et al., 2025).

Secara umum, arsitektur *Deep Learning* terdiri atas tiga lapisan utama, yaitu input layer, hidden layer, dan output layer. Input layer berfungsi menerima data masukan, hidden layer melakukan proses pembelajaran dan ekstraksi fitur, sedangkan output layer menghasilkan keluaran sesuai dengan tujuan model. Keunggulan utama *Deep Learning* terletak pada penggunaan banyak lapisan tersembunyi (*multiple hidden layers*) yang memungkinkan data diproses secara

bertahap dan hierarkis. Melalui proses tersebut, setiap lapisan akan menghasilkan representasi data yang semakin kompleks sehingga model mampu mengenali pola dan fitur yang lebih abstrak dengan tingkat akurasi yang lebih baik (Al-am et al., n.d.).

Dalam bidang *Computer Vision*, *Deep Learning* banyak digunakan untuk menyelesaikan permasalahan klasifikasi citra karena mampu mempelajari karakteristik visual suatu objek secara otomatis. Salah satu arsitektur *Deep Learning* yang paling populer untuk pengolahan citra adalah *Convolutional Neural Network* (CNN) (Prayoga et al., 2023). Arsitektur ini dirancang khusus untuk mengekstraksi fitur visual dari citra dan mengenali pola objek secara efektif, sehingga banyak dimanfaatkan dalam berbagai aplikasi klasifikasi dan deteksi objek.

2.5 *Convolutional Neural Network*

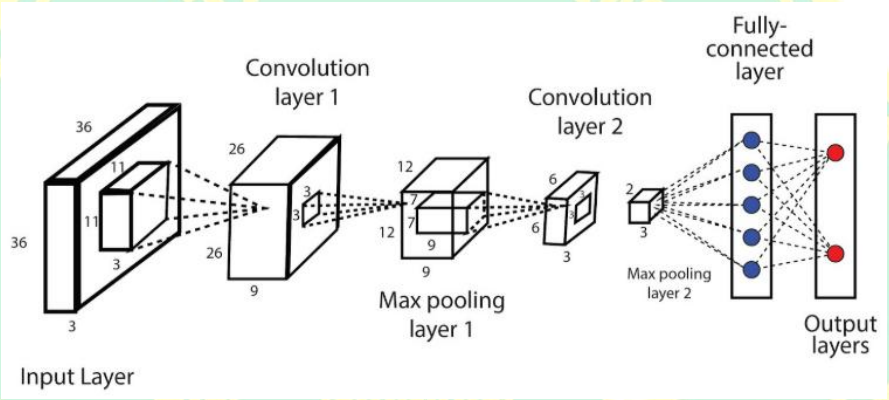
Convolutional Neural Network (CNN) merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang termasuk ke dalam jenis jaringan saraf tiruan feed forward, yaitu jaringan yang tidak memiliki umpan balik (*non-recurrent*). CNN dirancang secara khusus untuk memproses data berbentuk dua dimensi. Selain itu, CNN tergolong dalam *Deep Neural Network* karena memiliki struktur jaringan yang dalam, sehingga banyak digunakan dalam pengolahan dan analisis data citra (Nugroho et al., 2020).

CNN juga merupakan salah satu jenis *Artificial Neural Network* (ANN) yang difokuskan pada pemrosesan data seperti gambar, video, dan suara, di mana

arsitekturnya terdiri dari neuron-neuron yang memiliki bobot dan bias untuk memproses setiap input melalui operasi perkalian titik, sehingga dalam praktiknya CNN sering digunakan untuk mendeteksi dan mengenali objek pada data citra (Ibnul Rasidi et al., 2022).

2.5.1 Arsitektur *Convolutional Neural Network* (CNN)

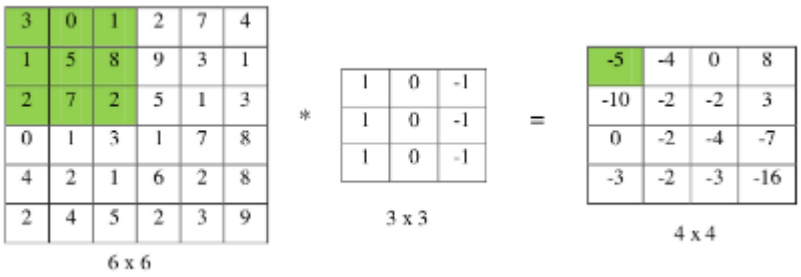
Terdapat tiga lapisan utama dalam proses pengolahan citra, yaitu *Convolution Layer, Pooling Layer* dan *Fully Connected Layer* (Ramadhani et al., 2023).



Gambar 2. 1 Arsitektur CNN (Dwivedi, 2021)

1. *Convolution Layer*

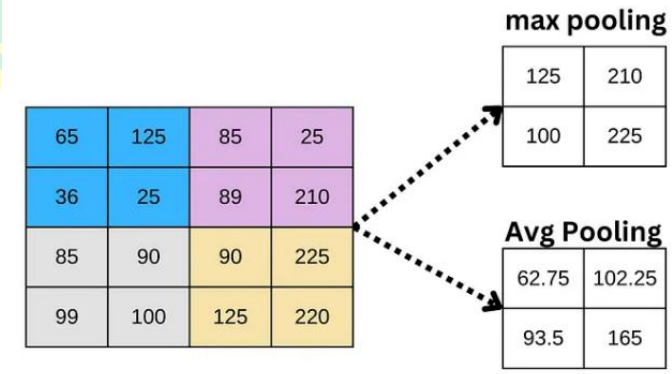
   Convolution Layer adalah komponen paling dasar dan utama yang membentuk jaringan CNN. Fungsi utamanya adalah mendeteksi dan mengekstrak fitur-fitur penting dari citra input secara otomatis melalui operasi matematika yang disebut konvolusi. Berbeda dengan model jaringan saraf biasa, lapisan ini mampu mengenali hubungan ruang (*spatial relationship*) antar piksel, sehingga sangat andal untuk mengenali bentuk atau objek pada gambar (Taye, 2023).



Gambar 2. 2 Proses *Convolution* (Rohim & Sari, 2021)

2. *Pooling Layer*

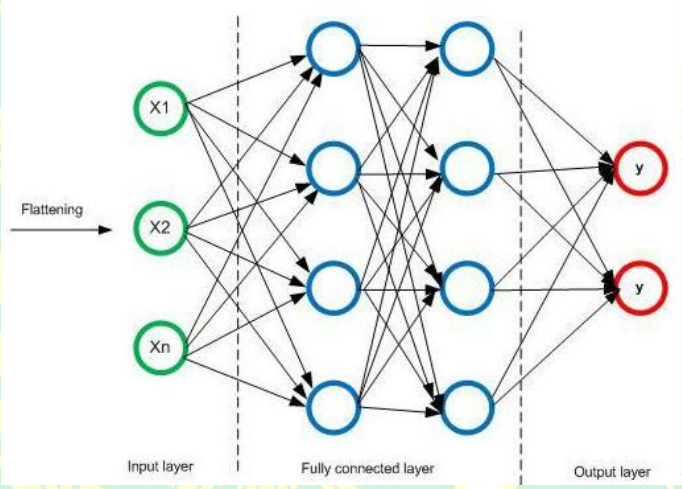
Pooling layer berfungsi untuk mereduksi ukuran sampel dimensi citra (*downsampling*) setelah melewati proses konvolusi, sehingga data menjadi lebih kecil, mudah dikelola oleh sistem, serta efektif dalam mencegah terjadinya *overfitting*. Dalam penerapannya, terdapat dua jenis *pooling* yang paling sering digunakan, yaitu *max pooling* yang bekerja dengan cara mengambil dan mengembalikan nilai piksel terbesar (maksimum) dari area gambar yang dicakup oleh kernel, serta *average pooling* yang berfungsi untuk menghitung dan mengembalikan nilai rata-rata dari seluruh bagian citra yang berada di dalam cakupan kernel tersebut (Nurul Khasanah, 2021).



Gambar 2. 3 *Proses Pooling* (Thida, 2024)

3. *Fully Connected Layer*

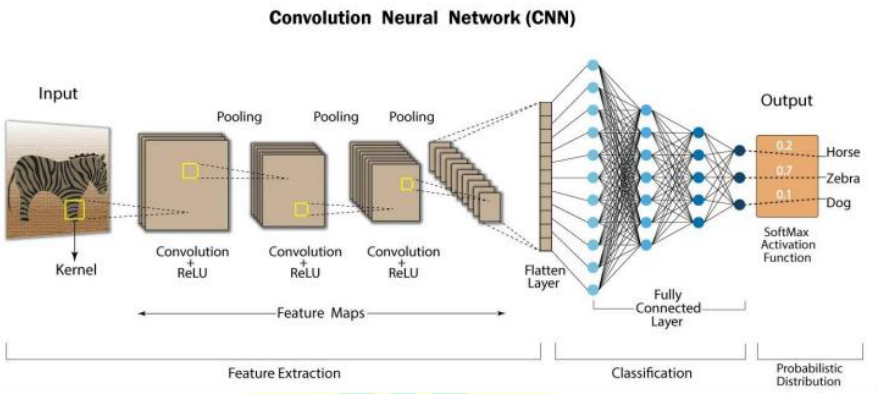
Fully Connected Layer merupakan lapisan di mana setiap neuron saling terhubung secara menyeluruh dengan seluruh neuron pada lapisan sebelum maupun sesudahnya. Pada tahap ini, data citra yang sebelumnya telah dikonversi ke dalam bentuk vektor satu dimensi akan dialirkan sebagai input utama untuk diproses lebih lanjut hingga menghasilkan sebuah prediksi keputusan akhir (Pane, 2024).



Gambar 2. 4 Proses *Fully Connected* (Herlambang, 2024)

2.5.2 Alur Kerja *Convolutional Neural Network* (CNN)

Setelah memahami fungsi dari setiap lapisan pada arsitektur CNN, proses klasifikasi citra dilakukan melalui serangkaian tahapan yang saling berhubungan.



Gambar 2. 5 Alur Kerja *Convolutional Neural Network* (Rebecca Jacob, 2025)

1. *Input image*: gambar dimasukkan ke jaringan.
2. *Convolution layer*: Ekstraksi fitur untuk menangkap pola seperti tepi, tekstur, atau bentuk.
3. *Activation Function* (ReLU): aktivasi agar model dapat mempelajari pola nonlinier.
4. *Pooling layer*: ukuran data diperkecil supaya komputasi lebih efisien dan fitur penting tetap tersimpan.
5. *Flatten*: hasil fitur peta diubah menjadi vektor 1 dimensi.
6. *Fully connected layer*: Menghubungkan seluruh fitur.
7. *SoftMax:* Menghitung probabilitas tiap kelas.
8. *Output*: hasil klasifikasi, misalnya organik atau anorganik.

Berdasarkan prinsip kerjanya, CNN dapat mengekstraksi fitur dan mengklasifikasikan citra secara otomatis melalui beberapa lapisan yang saling terhubung. Namun, CNN konvensional membutuhkan parameter dan komputasi yang relatif besar sehingga kurang sesuai untuk perangkat dengan kemampuan komputasi terbatas. Oleh karena itu, dikembangkan arsitektur MobileNetV2 yang mempertahankan prinsip dasar CNN dengan menerapkan *Depthwise Separable*

Convolution, *Linear Bottleneck*, dan *Inverted Residual* untuk meningkatkan efisiensi komputasi tanpa mengurangi performa klasifikasi secara signifikan.

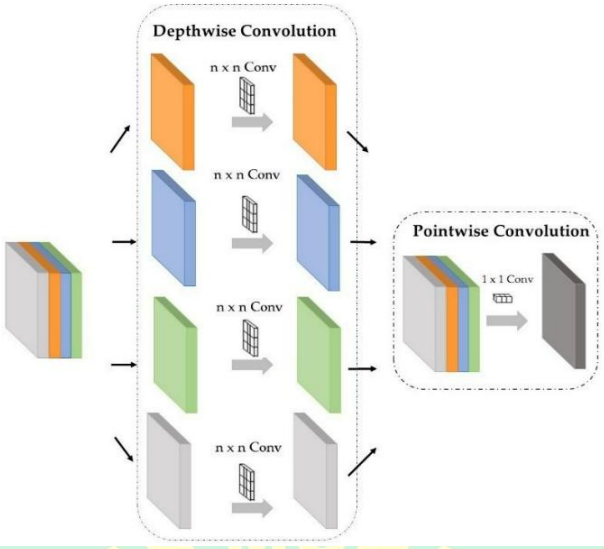
2.6 Arsitektur MobileNetV2

MobileNetV2 merupakan salah satu arsitektur *Convolutional Neural Network* (CNN) yang dikembangkan untuk menghasilkan model dengan ukuran yang lebih ringan dan efisien tanpa mengurangi kemampuan akurasi dalam melakukan klasifikasi citra. Sebagai pengembangan dari MobileNetV1, arsitektur ini sangat sesuai untuk diterapkan pada perangkat dengan kapasitas komputasi yang terbatas maupun pada aplikasi yang memerlukan pemrosesan citra secara langsung atau *real-time* (Alfan & Kumalasari, 2026).

Efisiensi dan peningkatan kinerja pada MobileNetV2 dicapai berkat integrasi tiga mekanisme utama dalam arsitekturnya, yaitu *Depthwise Separable Convolution*, *Inverted Residual*, dan *Linear Bottleneck*.

1. *Depthwise Separable Convolution* (Pemisahan Konvolusi)

   Depthwise separable convolution merupakan fitur berupa blok *deep learning* pada model MobileNetV1 yang mereduksi daya komputasi dan ukuran model dengan cara memecah konvolusi standar yang berat menjadi dua tahap mandiri, yaitu *depthwise convolution* untuk menyaring visual pada tiap saluran input secara terpisah, dan *pointwise convolution* menggunakan kernel 1x1 untuk mengombinasikan hasilnya secara linear menjadi saluran baru (Baay et al., 2021).



Gambar 2. 6 Proses *Depthwise Separable Convolution* (Lu, 2022)

Untuk menunjukkan efisiensi Depthwise Separable Convolution dibandingkan konvolusi standar, dapat dilakukan perbandingan jumlah operasi komputasi. Pada konvolusi standar, jumlah operasi dihitung menggunakan persamaan:

$$D_K \times D_K \times M \times N \times D_F \times D_F$$

Sedangkan pada *Depthwise Separable Convolution*, proses konvolusi dibagi menjadi dua tahap, yaitu *depthwise convolution* dengan jumlah operasi:

$$D_K \times D_K \times M \times D_F \times D_F$$

dan *pointwise convolution* dengan jumlah operasi:

$$M \times N \times D_F \times D_F$$

Sehingga total jumlah operasi pada Depthwise Separable Convolution menjadi:

$$D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F$$

Berdasarkan persamaan tersebut, *Depthwise Separable Convolution* membutuhkan jumlah operasi yang lebih sedikit dibandingkan konvolusi

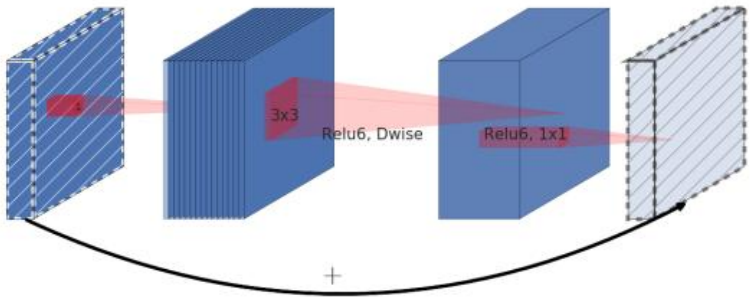
standar karena proses konvolusi dipisahkan menjadi dua tahap, sehingga MobileNetV2 menjadi lebih ringan dan efisien tanpa mengurangi kemampuan dalam mengekstraksi fitur citra.

Keterangan:

- D_K= ukuran kernel konvolusi.
- M= jumlah channel masukan (*input feature map*).
- N= jumlah filter atau channel keluaran.
- D_F= ukuran *feature map*.

2. *Inverted Residual* (Ekspansi dan Proyeksi Terbalik)

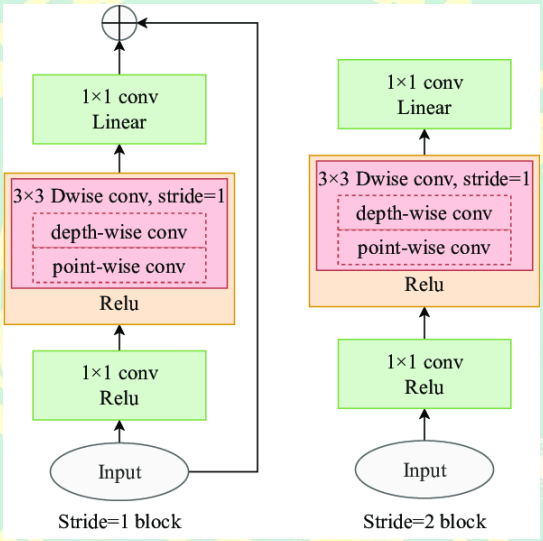
   Inverted Residual merupakan sebuah struktur blok penyusun pada jaringan saraf yang membalikkan konsep blok residu konvensional, di mana jalur pintas (*shortcut*) justru menghubungkan antar-lapisan bermembran tipis atau berdimensi rendah. Di dalam blok ini, data input yang awalnya tipis akan diperluas terlebih dahulu ke dimensi yang jauh lebih tebal melalui lapisan ekspansi agar jaringan dapat menangkap detail fitur gambar secara maksimal, sebelum akhirnya dikompresi kembali ke ukuran semula di akhir blok (Sandler et al., n.d.).



Gambar 2. 7 Proses *Inverted Residual* (Sandler et al., n.d.)

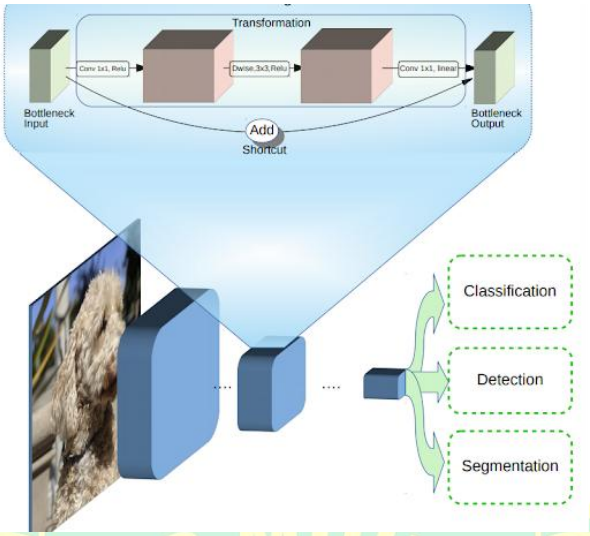
3. *Linear Bottleneck* (Pertahanan Informasi lewat Aktivasi Linear)

   Linear Bottleneck merupakan suatu teknik penataan lapisan di bagian akhir blok residual terbalik yang sengaja mengganti fungsi aktivasi non-linear seperti ReLU dengan operasi linear biasa, di mana penghapusan sifat non-linear pada lapisan berdimensi rendah ini bertujuan untuk mereduksi dimensi fitur secara jauh lebih efisien sekaligus mencegah hilangnya data visual yang halus, sehingga daya representasi dan informasi penting pada model tetap terjaga dengan utuh saat mengenali objek (Sardika & Widhiarso, 2025).



Gambar 2. 8 Diagram *skematik blok residual dengan bottleneck linier* (Model et al., 2022)

Selain itu, MobileNetV2 juga tersedia sebagai *pre-trained* model yang telah dilatih menggunakan dataset berskala besar, seperti *ImageNet*. Kondisi ini memungkinkan penerapan *Transfer Learning*, sehingga proses pelatihan menjadi lebih efisien karena model tidak perlu dibangun dan dilatih dari awal (Pane, 2024). Arsitektur MobileNetV2 ditunjukkan pada Gambar berikut:



Gambar 2. 9 Arsitektur MobileNetV2 (Marpaung et al., 2023)

Proses klasifikasi citra dilakukan melalui beberapa tahapan sebagai berikut.

1. Input Image (224 × 224 × 3)

   Citra masukan berukuran 224 × 224 piksel dengan tiga kanal warna (RGB) diterima sebagai input model.

2. Convolution 3 × 3

   Tahap ini bertujuan mengekstraksi fitur-fitur dasar pada citra, seperti tepi, tekstur, dan pola sederhana.

3. 17 Bottleneck Blocks (Inverted Residual)

   Fitur diproses menggunakan kombinasi Expansion Layer, Depthwise Convolution, Linear Bottleneck, dan Inverted Residual untuk menghasilkan representasi fitur yang lebih kompleks dengan jumlah parameter dan komputasi yang lebih efisien.

4. Convolution 1 × 1

   Layer ini berfungsi menggabungkan serta memperkaya informasi fitur yang dihasilkan dari bottleneck blocks.

5. *Global Average Pooling*

   Feature map diringkas dengan menghitung nilai rata-rata pada setiap kanal sehingga dimensi fitur dan jumlah parameter dapat dikurangi.

6. *Fully Connected Layer*

   Fitur hasil ekstraksi diolah menjadi skor yang akan digunakan sebagai dasar dalam proses klasifikasi.

7. Softmax

   Skor klasifikasi diubah menjadi nilai probabilitas untuk masing-masing kelas sehingga model dapat menentukan kelas yang paling sesuai.

8. Output

   Tahap akhir menghasilkan prediksi kategori sampah, yaitu organik atau non-organik, berdasarkan nilai probabilitas tertinggi.

2.7 *Transfer Learning*

Transfer Learning yang juga dikenal sebagai pre-trained model, merupakan pendekatan dalam *Deep Learning* yang memanfaatkan model CNN yang telah dilatih sebelumnya (*pre-trained model*) menggunakan suatu dataset untuk diterapkan pada tugas atau dataset yang berbeda. Pendekatan ini memungkinkan model memanfaatkan pengetahuan yang telah diperoleh selama proses pelatihan awal sehingga dapat mempercepat proses pengembangan dan meningkatkan efisiensi pelatihan. Dibandingkan dengan CNN sederhana, model *Transfer Learning* umumnya memiliki arsitektur yang lebih dalam dan kompleks. Beberapa arsitektur CNN yang umum digunakan dalam *Transfer Learning* meliputi

VGG16, InceptionV3, ResNet50V2, dan MobileNetV2. Model-model tersebut telah dilatih menggunakan dataset berskala besar, seperti ImageNet, sehingga mampu mengenali berbagai fitur visual yang dapat dimanfaatkan kembali pada berbagai tugas klasifikasi citra (Rozaqi et al., 2021).

Penerapan *Transfer Learning* menawarkan berbagai keuntungan, seperti mempercepat proses pelatihan model, mengurangi kebutuhan data pelatihan dalam jumlah besar, serta meningkatkan performa dan akurasi model dibandingkan dengan pelatihan yang dilakukan dari awal (*training from scratch*). Dengan memanfaatkan pengetahuan yang telah dipelajari dari dataset sebelumnya, model dapat beradaptasi lebih cepat terhadap tugas baru (Mufidatuzzainiya & Faisal, 2025). Oleh karena itu, *Transfer Learning* banyak diterapkan dalam berbagai aplikasi *Computer Vision* karena mampu menghasilkan model yang efektif dan akurat dengan penggunaan waktu serta sumber daya komputasi yang lebih efisien.

2.8 *Image Pre-processing*

Image pre-processing merupakan tahapan awal dalam pengolahan citra yang dilakukan untuk mempersiapkan data sebelum digunakan pada proses pelatihan model *Deep Learning*. Tahap ini bertujuan menghasilkan data citra yang memiliki ukuran, format, dan rentang nilai piksel yang seragam sehingga proses pembelajaran model dapat berlangsung secara lebih optimal. Selain itu, *image pre-processing* juga berfungsi untuk meningkatkan kualitas data, memperbaiki kemampuan generalisasi model, serta mengurangi risiko terjadinya *overfitting* (Azzahra, 2024).

Pada penelitian ini, tahapan image pre-processing yang diterapkan meliputi resize, normalisasi, dan augmentasi data.

1. Resize

   Resize merupakan proses mengubah ukuran citra menjadi dimensi yang seragam agar sesuai dengan kebutuhan model. Tahap ini bertujuan untuk memastikan seluruh citra memiliki ukuran input yang konsisten sehingga dapat diproses secara optimal oleh model. Meskipun arsitektur MobileNetV2 mendukung berbagai ukuran input di atas 32 × 32 piksel, penelitian ini menggunakan ukuran standar 224 × 224 piksel sesuai dengan ukuran masukan bawaan (default) MobileNetV2 (Tinggi et al., 2020). Pemilihan ukuran tersebut dilakukan karena mampu memberikan keseimbangan antara kualitas informasi citra dan efisiensi komputasi selama proses pelatihan maupun klasifikasi.

2. *Normalization*

   Normalisasi merupakan proses mengubah rentang nilai piksel citra menjadi skala yang lebih seragam agar sesuai dengan kebutuhan model. Pada penelitian ini, nilai piksel citra yang semula berada pada rentang 0–255 dinormalisasi menjadi 0–1 dengan membagi setiap nilai piksel menggunakan nilai maksimum, yaitu 255. Proses ini bertujuan untuk mempercepat pelatihan, meningkatkan stabilitas model, dan membantu model mempelajari pola data dengan lebih efektif (Rifka Alkhilyatul Ma'rifat, I Made Suraharta, 2024).

   Persamaan normalisasi yang digunakan adalah:

$$\text{rescale} = \frac{\text{img}}{255}$$

dengan keterangan:

a. rescale = nilai piksel setelah normalisasi.

b. img = nilai piksel citra sebelum normalisasi.

c. 255 = nilai maksimum piksel pada citra 8-bit.

3. Augmentation Data

Augmentation data adalah teknik yang digunakan untuk memperbanyak variasi data pelatihan melalui penerapan transformasi tertentu pada citra. Teknik ini bertujuan meningkatkan keberagaman dataset sehingga model dapat mempelajari karakteristik objek secara lebih optimal, meningkatkan kemampuan generalisasi, dan mengurangi risiko *overfitting* (Bintang & Azhar, 2024).

2.9 Klasifikasi Citra

Klasifikasi citra merupakan proses pengelompokan atau pemberian label terhadap citra berdasarkan karakteristik visual yang dimilikinya ke dalam kelas-kelas tertentu. Proses ini dilakukan dengan menganalisis informasi yang terkandung pada piksel citra sehingga setiap kelas dapat merepresentasikan objek atau entitas dengan ciri khas yang berbeda. Tujuan utama klasifikasi citra adalah mengidentifikasi dan membedakan objek yang terdapat dalam citra, sehingga setiap objek dapat dikategorikan sesuai dengan kelas yang telah ditentukan (Ii & Pearson, 2021).

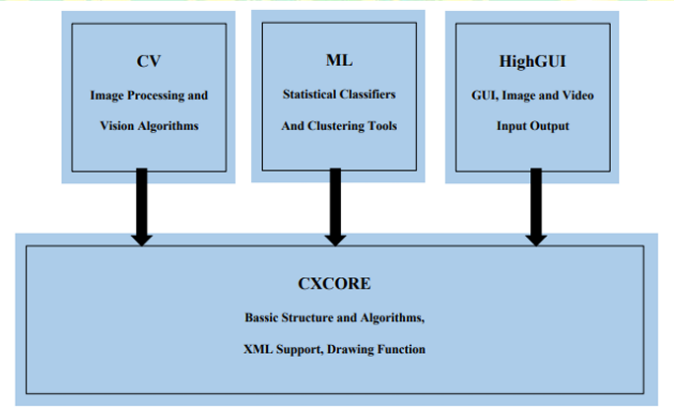
Secara umum, klasifikasi citra dapat dibedakan menjadi dua pendekatan utama, yaitu *supervised classification* (klasifikasi terawasi) dan *unsupervised*

classification (klasifikasi tidak terawasi). Di antara kedua pendekatan tersebut, supervised classification lebih banyak diterapkan karena mampu memberikan hasil klasifikasi yang lebih akurat dan konsisten. Metode ini memanfaatkan data latih yang telah memiliki label atau kategori sebagai dasar pembelajaran model. Melalui data tersebut, model dapat mempelajari karakteristik dari setiap kelas sehingga mampu mengidentifikasi dan mengelompokkan data baru ke dalam kelas yang sesuai (Multispektral, 2022).

2.10 OpenCV

OpenCV (*Open Source Computer Vision Library*) merupakan pustaka perangkat lunak sumber terbuka yang dikembangkan untuk mendukung berbagai aplikasi *Computer Vision* dan *Machine Learning*. OpenCV menyediakan infrastruktur yang komprehensif untuk pengolahan citra dan video, sehingga dapat membantu mempercepat pengembangan sistem berbasis penglihatan komputer pada berbagai kebutuhan, termasuk aplikasi komersial. Dengan lisensi BSD yang bersifat terbuka, OpenCV memungkinkan pengguna untuk memanfaatkan, memodifikasi, dan mendistribusikan kode sesuai kebutuhan. Selain itu, OpenCV menyediakan lebih dari 2.500 algoritma yang telah dioptimalkan, mencakup berbagai teknik *Computer Vision* dan *Machine Learning*, mulai dari metode klasik hingga metode modern. Kelengkapan fitur serta kinerjanya yang efisien menjadikan OpenCV sebagai salah satu pustaka yang paling banyak digunakan dalam pengembangan aplikasi Computer Vision (Muchtar & Apriadi, n.d.).

OpenCV tersusun atas beberapa library utama yang mendukung berbagai kebutuhan dalam pengembangan aplikasi *Computer Vision*. Library tersebut meliputi *Computer Vision* (CV) yang menyediakan berbagai algoritma pengolahan citra dan analisis visual, *Machine Learning* (ML) untuk mendukung penerapan algoritma pembelajaran mesin, HighGUI yang berfungsi sebagai antarmuka grafis untuk menampilkan citra dan video, serta Image and Video I/O yang digunakan untuk proses pembacaan dan penyimpanan data citra maupun video. Selain itu, OpenCV juga memiliki CXCORE yang menyediakan struktur data dasar, dukungan format XML, serta berbagai fungsi grafis, dan CvAux yang berisi fungsi-fungsi pendukung untuk melengkapi kebutuhan pengolahan citra dan *Computer Vision*.



Gambar 2. 10 Struktur dan Konten OpenCV (Isum et al., 2021)

2.11 *Confusion Matrix*

Confusion matrix merupakan metode evaluasi yang digunakan untuk membandingkan hasil prediksi model dengan data sebenarnya. Melalui perbandingan tersebut, *confusion matrix* dapat memberikan gambaran mengenai

tingkat kinerja dan akurasi model klasifikasi secara lebih intuitif, termasuk kemampuan model dalam mengklasifikasikan data ke dalam kelas yang benar maupun kesalahan klasifikasi yang terjadi (Nurul Khasanah, 2021). Terdapat beberapa istilah umum yang dapat dipakai dalam proses pengukuran kinerja dari model klasifikasi yatu:

1. *True Positive* (TP) merupakan data positif yang diprediksi benar
2. *True Negative* (TN) merupakan data negatif yang diprediksi benar
3. *False Positive* (FP) merupakan data negatif namun diprediksi sebagai data positif
4. *False Negative* (FN) merupakan data positif namun diprediksi sebagai data negatif

Berikut tabel perhitungan *confusion matrix* (Informatika et al., 2025):

Tabel 2.1 *confusion matrix*

		Predicted Class		
				Total
Actual Class	Yes	TP	FN	P
	No	FP	TN	N
	Total	P	N	P+N

Confusion Matrix membantu dalam evaluasi metrik seperti *Accuracy*, *Precisi on*, *Recall*, dan *F1-Score* , yang penting untuk menilai kinerja model dalam memprediksi data uji. Berikut rumusnya:

a. Accuracy

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

b. Precision

$$Precision = \frac{TP}{TP + FP} \times 100\%$$

c. *Recall*

$$Recall = \frac{TP}{TP + FN} \times 100\%$$

d. *F1-Score*

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$