

BAB 2

LANDASAN TEORI

2.1 Tuberkulosis

Tuberkulosis (TBC) adalah gangguan kesehatan akibat infeksi yang dapat ditularkan dan diakibatkan karena bakteri *Mycobacterium tuberculosis* (MTB). Meskipun organ paru adalah bagian yang paling umum diserang, MTB berpotensi memicu infeksi pada hampir seluruh bagian tubuh manusia. Penyebaran TBC dapat terjadi secara *airborne*, yaitu saat penderita TBC batuk, bersin, dan berbincang, kemudian mengeluarkan *droplet* kecil dengan bakteri MTB ke udara [21]. Terdapat dua kondisi klinis utama yang perlu dipahami dalam konteks TBC, yaitu infeksi laten dan TBC aktif. Infeksi MTB akan berkembang melalui kondisi terkontrol pada tubuh, dimana bakteri terisolasi pada bagian granuloma (TBC laten) hingga berkembang menjadi TBC paru aktif yang bersifat menular [22].

Bakteri MTB dapat bertahan dalam kondisi dorman dalam kurun waktu yang panjang dan menetap di dalam tubuh tanpa menunjukkan tanda-tanda penyakit, sehingga banyak orang menjadi pembawa asimtomatik atau dikenal sebagai TBC inaktif [22]. Dari sisi patogenisitas, MTB dianggap sebagai salah satu patogen manusia yang paling adaptif. Bakteri ini memiliki sejumlah faktor virulensi yang membantunya menghindari pertahanan mukosa tubuh. Ketika sistem imun tubuh berhasil menekan infeksi, bakteri akan tetap berada dalam bentuk laten. Tetapi saat sistem imun melemah, infeksi ini berpotensi menjadi TBC aktif yang menular [23].

Secara radiologis, kasus TBC pada citra *X-ray* paru-paru ditandai dengan adanya pola retikular dan nodular bilateral yang menunjukkan penyebaran lesi pada jaringan paru, terutama lebih dominan di zona tengah dan bawah paru kanan. Selain itu, terdapat *micronodular interstitial* yang mengindikasikan penyebaran infeksi secara luas pada jaringan interstitial, disertai kemungkinan efusi sebagai respons inflamasi. Temuan lain meliputi bronkiektasis pada apeks paru kiri di area lingula yang mencerminkan kerusakan kronis saluran napas akibat proses infeksi granulomatosa, serta penebalan *pleura apical* [24]. Gambar X-ray pada penyakit TBC ditunjukkan oleh Gambar 2.1.



Gambar 2.1. Contoh Gambar X-ray Tuberkulosis
Sumber: [25]

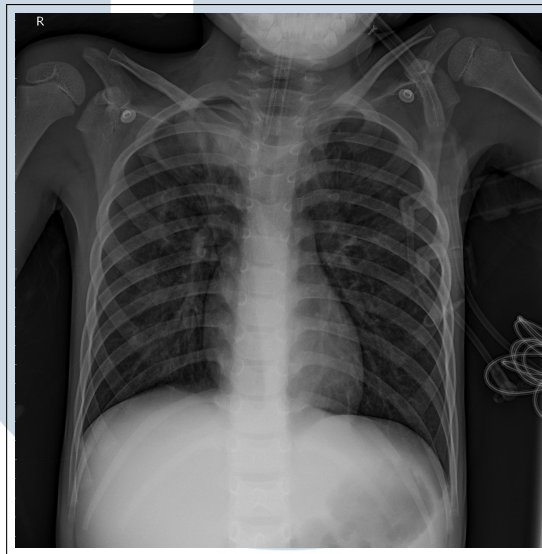
2.2 Pneumonia

Pneumonia adalah gangguan infeksi di sistem pernapasan yang umumnya terjadi akibat proses infeksi pada alveolus serta bagian distal saluran napas paru-paru. Penyakit pneumonia menjadi salah satu isu kesehatan penting, karena berhubungan dengan tingginya jumlah kasus dan resiko kematian, baik dalam kurun waktu pendek maupun panjang, pada berbagai usia pada seluruh dunia. [14].

Berdasarkan lokasi penularan, pneumonia diklasifikasikan menjadi dua kategori utama, yaitu *Hospital Acquired Pneumonia* atau pneumonia yang ditularkan melalui tempat fasilitas kesehatan seperti rumah sakit dan *Community-Acquired Pneumonia* atau pneumonia yang didapatkan diluar dari tempat fasilitas kesehatan. Klasifikasi ini penting karena menentukan perbedaan profil patogen penyebab, faktor risiko, serta pendekatan terapeutik yang digunakan [14].

Dari sisi etiologi, berbagai jenis mikroorganisme dapat menyebabkan pneumonia, termasuk bakteri, virus respirasi, dan *fungi*, dengan prevalensi yang bervariasi secara geografis. Pneumonia bakteri ditandai oleh peradangan pada parenkim paru dan ruang alveolar. Berdasarkan manifestasi klinisnya, pengidap pneumonia dapat menunjukkan keluhan berupa peningkatan suhu tubuh, batuk, kesulitan bernapas, serta rasa sakit pada dada. Kondisi ini juga berisiko menimbulkan komplikasi seperti, pneumonia nekrotikans, empiema, meningitis, sepsis, hingga gangguan fungsi pada beberapa organ. [26].

Dari aspek pencitraan medis, kasus pneumonia umumnya ditandai dengan munculnya radio-opasitas atau area berwarna putih pada paru-paru, terutama di bagian alveoli. Kondisi ini menunjukkan adanya akumulasi cairan inflamasi (eksudat) yang mengisi ruang udara, sehingga menyebabkan peningkatan densitas pada area tersebut dan mengindikasikan proses infeksi pada jaringan paru [27]. Gambar X-ray pada penyakit pneumonia ditunjukkan oleh Gambar 2.2.

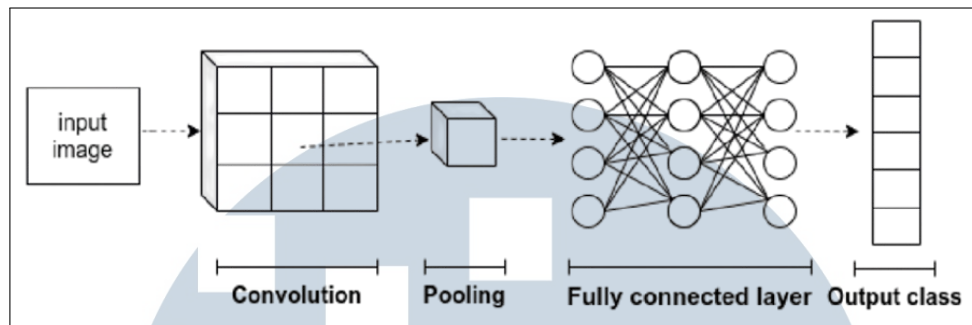


Gambar 2.2. Contoh Gambar X-ray Pneumonia

Sumber: [25]

2.3 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan algoritma *deep learning* yang dikembangkan dari *Artificial Neural Network* (ANN) dan dibangun secara spesifik sebagai pengolah data dalam bentuk grid seperti citra digital, mekanisme CNN dibuat untuk menyerupai sistem optik manusia, di mana setiap neuron hanya memproses bagian tertentu dari input yang disebut *receptive field*, sehingga memungkinkan proses *feature extraction* dilakukan pada tingkat lokal secara bertahap. CNN terdiri dari beberapa lapisan utama, yaitu *convolutional layer* yang berfungsi mengekstraksi fitur seperti tepi, tekstur, dan pola, *activation layer* seperti *Rectified Linear Unit* untuk menambahkan kemampuan model dalam mempelajari pola non-linear, *pooling layer* yang berfungsi untuk mengurangi dimensi data dengan tetap mempertahankan informasi penting, serta *fully connected layer* yang digunakan sebagai proses klasifikasi pada fitur yang telah dipelajari [28]. Alur kerja CNN diperlihatkan oleh Gambar 2.3.



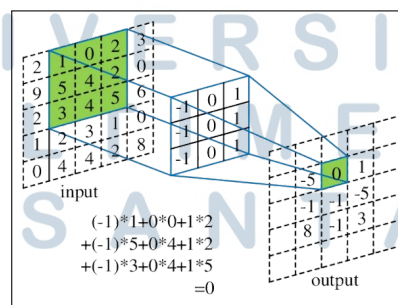
Gambar 2.3. Struktur CNN

Sumber: [29]

2.3.1 Convolution Layer

Convolutional layer adalah komponen utama pada arsitektur *Convolutional Neural Network* (CNN) yang bertugas sebagai *feature extraction* atau mengekstraksi fitur spasial pada data input. Layer ini terdiri dari sejumlah filter yang digunakan untuk mendeteksi pola tertentu pada citra, seperti tekstur, tepi, maupun bentuk objek. Tiap kernel memiliki parameter berupa bobot yang dipelajari secara otomatis selama proses pelatihan sehingga mampu mengenali fitur yang relevan [30].

Dalam prosesnya, citra input direpresentasikan dalam bentuk matriks, kemudian filter berukuran lebih kecil akan digeser secara horizontal dan vertikal ke seluruh dimensi spasial data masukan menggunakan parameter *stride* dan *padding*. Pada setiap posisi, kernel akan berinteraksi dengan bagian citra yang dilaluinya dan menghasilkan satu nilai. Proses ini dilakukan secara berulang hingga seluruh area citra terproses, sehingga terbentuk suatu *feature map* sebagai representasi hasil *feature extraction* dari citra tersebut [28, 30, 31]. Gambar 2.4 menunjukkan cara kerja *Convolution Layer*.



Gambar 2.4. Convolution Layer

Sumber: [32]

2.3.2 Activation Function

Activation function merupakan komponen penting pada arsitektur *deep learning* yang berfungsi sebagai penentu apakah suatu neuron akan meneruskan informasi ke lapisan berikutnya atau tidak . Fungsi ini mengolah output dari neuron menjadi nilai tertentu yang akan digunakan sebagai input pada neuron selanjutnya. Peran utama *activation function* adalah membantu model dalam mengenali hubungan dan karakteristik data yang lebih rumit. Apabila *activation function* tidak digunakan, jaringan neuron hanya bekerja secara linear, sehingga tidak mampu menyelesaikan permasalahan yang kompleks [33].

Selain itu, *activation function* juga berkontribusi dalam meningkatkan performa model, seperti membantu proses pembelajaran berjalan lebih optimal, meningkatkan akurasi, serta mengurangi risiko *overfitting* . Dalam implementasinya, *activation function* umumnya bersifat non-linear dan terbagi menjadi dua jenis, yaitu fungsi dengan parameter tetap dan fungsi yang parameternya dapat dipelajari. Beberapa contoh *activation function* yang sering digunakan yaitu sigmoid, tanh, dan ReLU, yang masing-masing memiliki karakteristik berbeda sesuai dengan kebutuhan model *deep learning* [33].

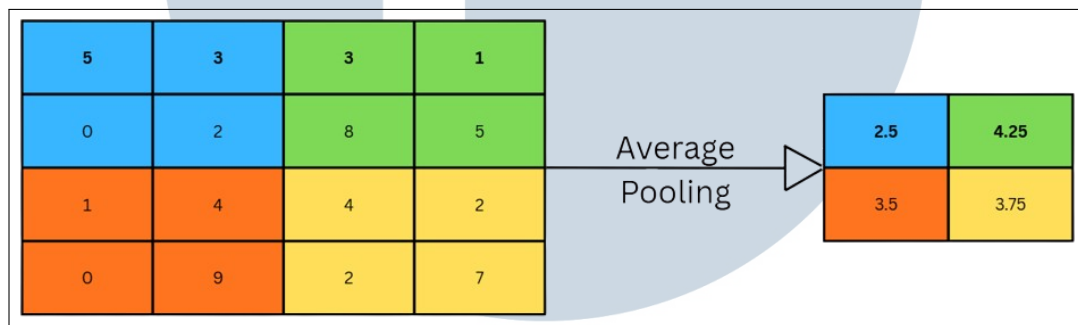
2.3.3 Pooling Layer

Pooling layer merupakan proses pada *Convolutional Neural Network* yang berfungsi sebagai *down-sampling* pada *feature map*, serta menangkap fitur citra dalam skala yang lebih besar yang bersifat invarian terhadap transformasi lokal kecil, seperti translasi, skala, dan rotasi. Proses ini dilakukan dengan cara mengagregasi nilai-nilai dari setiap wilayah spasial tertentu, misalnya dengan menghitung rata-rata nilai pada setiap area berukuran tertentu (misalnya 3×3) pada setiap fitur [34].

Selain itu, *pooling* juga berperan dalam memperluas *receptive field* dari neuron pada lapisan konvolusi di lapisan berikutnya, sehingga model dapat memahami konteks fitur yang lebih luas. Penggunaan *pooling layer* turut membantu mengurangi kompleksitas komputasi dan kebutuhan memori karena ukuran resolusi *feature map* menjadi lebih kecil, tanpa menghilangkan informasi penting yang diperlukan untuk proses selanjutnya. Dalam konteks analisis citra medis, *pooling* juga berkontribusi dalam menangani variasi ukuran dan posisi lesi pada citra [34].

A Average Pooling

Average pooling merupakan metode pada *pooling layer* yang bekerja dengan menghitung nilai *average* pada setiap elemen dalam suatu kernel pada *feature map*. Proses ini dilakukan dengan menggeser kernel berukuran tertentu (misalnya 2×2 atau 3×3) ke seluruh bagian *feature map*, kemudian setiap nilai dalam kernel tersebut dijumlahkan dan dibagi dengan jumlah elemen untuk menghasilkan satu nilai. Hasil yang diberikan prosedur ini merupakan *feature map* baru dengan ukuran yang lebih kecil, di mana setiap nilai merepresentasikan rata-rata dari wilayah lokal pada *feature map* sebelumnya [34]. Gambar 2.5 adalah ilustrasi cara kerja *average pooling*.

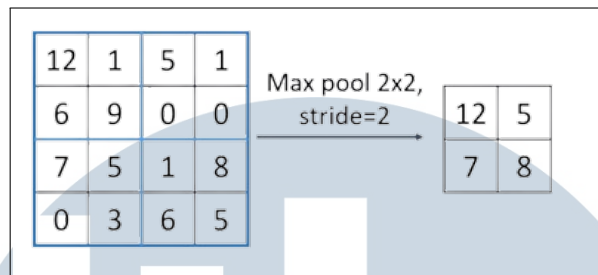


Gambar 2.5. Average Pooling Layer

Sumber: [35]

B Max Pooling

Max pooling merupakan metode *pooling* yang mengambil nilai paling tinggi untuk semua kernel yang ada di *feature map*. Metode ini banyak digunakan karena mampu mempertahankan fitur yang paling menonjol atau dominan dalam suatu area, sehingga lebih efektif dalam proses ekstraksi fitur penting. *Max pooling* merupakan salah satu teknik yang banyak dimanfaatkan pada sebagian besar aplikasi *computer vision* karena kemampuannya dalam meningkatkan performa model dengan tetap menjaga informasi utama dari data [34]. Cara kerja *max pooling* diperlihatkan di Gambar 2.6.

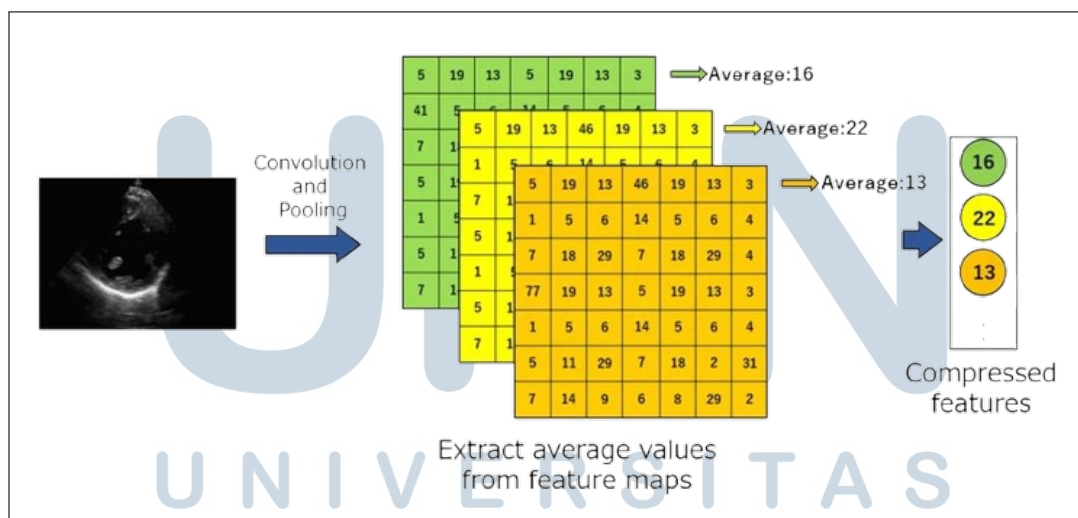


Gambar 2.6. Max Pooling Layer

Sumber: [34]

C Global Average Pooling

Global Average Pooling (GAP) adalah metode *global pooling* yang digunakan untuk merangkum *feature map* dengan cara mengkalkulasi rata-rata semua nilai aktivasi pada setiap *channel*. Tahapan tersebut memberikan satu nilai representatif untuk semua *feature map* yang mencerminkan informasi secara keseluruhan. Dengan demikian, GAP mengubah *feature map* berdimensi spasial menjadi representasi yang lebih sederhana tanpa mempertahankan struktur spasialnya [36]. Gambar 2.7 adalah ilustrasi cara kerja *global average pooling*.



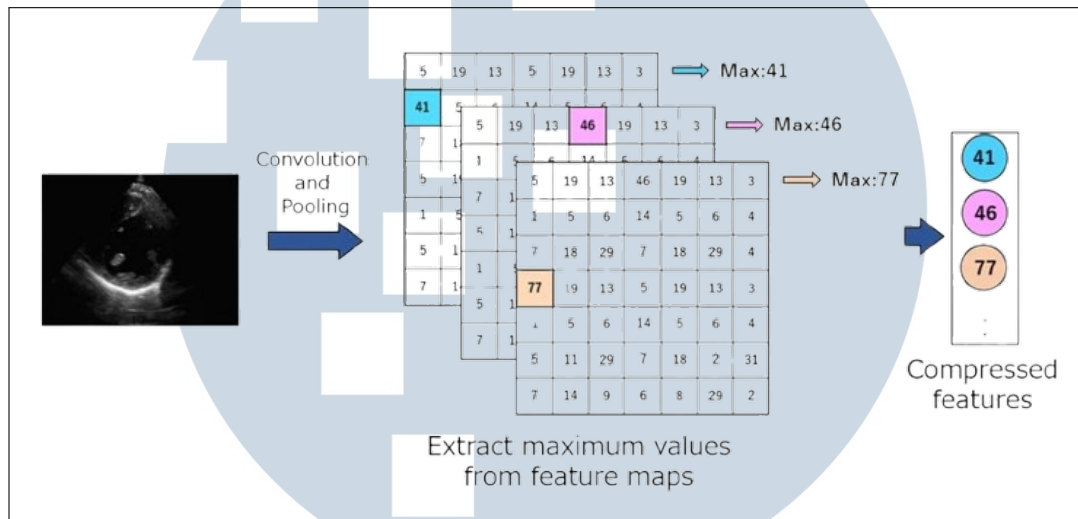
Gambar 2.7. Global Average Pooling

Sumber: [37]

D Global Max Pooling

Global Max Pooling (GMP) adalah teknik *global pooling* yang menggunakan nilai aktivasi maksimal pada setiap *feature map* sebagai representasi

akhir. Dalam proses ini, hanya nilai maksimum yang diambil dari keseluruhan area *feature map*, sehingga setiap *channel* direduksi menjadi satu nilai tunggal. Metode ini juga mengubah *feature map* menjadi bentuk yang lebih ringkas dengan menghilangkan dimensi spasial [36]. Gambar 2.8 adalah ilustrasi cara kerja *global max pooling*.

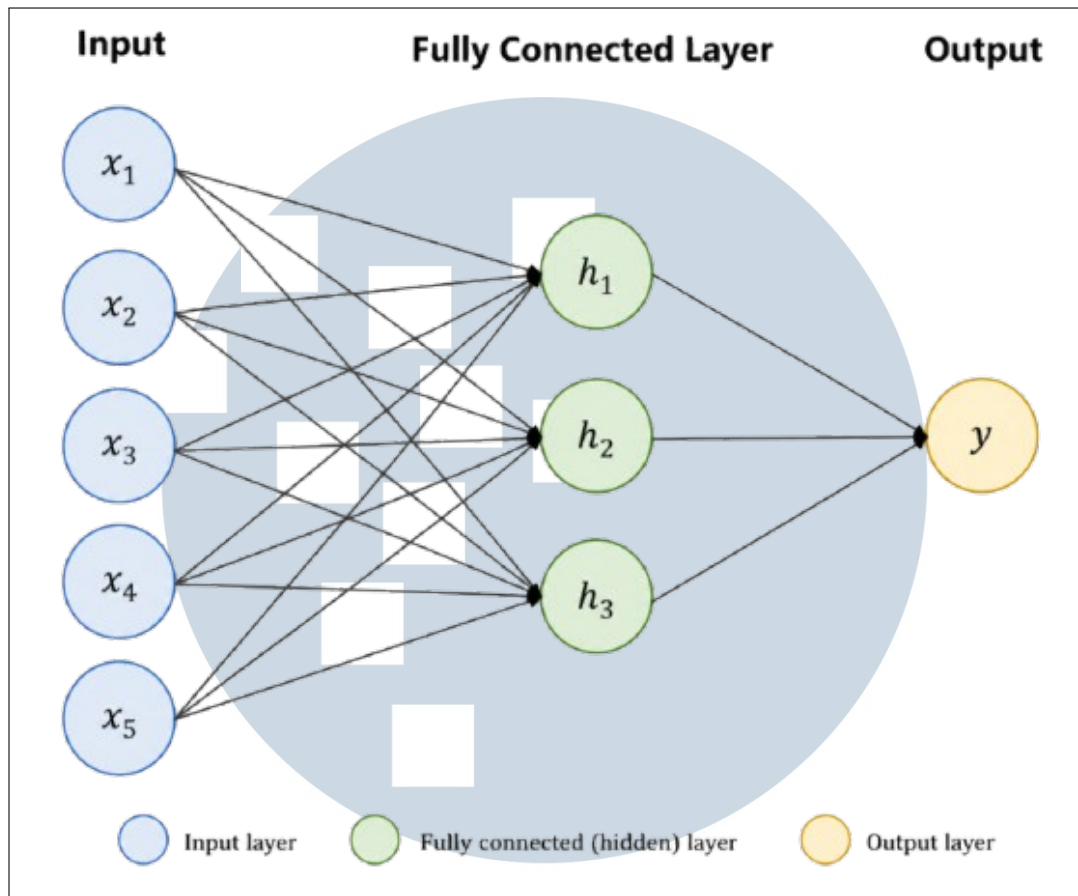


Gambar 2.8. Global Max Pooling

Sumber: [37]

2.3.4 Fully Connected Layer

Fully Connected Layer (FCL) merupakan lapisan yang menghubungkan setiap neuron dengan seluruh neuron pada lapisan sebelumnya. Lapisan ini berfungsi untuk mengubah hasil *feature extraction* pada layer sebelumnya menjadi representasi akhir yang digunakan dalam proses klasifikasi. Dalam CNN, *Fully Connected layer* biasanya terletak pada bagian akhir jaringan dan berfungsi untuk mengambil keputusan dengan mengolah fitur menjadi output berupa probabilitas kelas melalui fungsi aktivasi [38, 39]. struktur FCL diperlihatkan di Gambar 2.9.



Gambar 2.9. Fully Connected Layer

Sumber: [39]

2.3.5 Dropout

Dropout merupakan teknik yang dilakukan dengan mengabaikan sebagian aktivasi neuron selama proses pelatihan pada *Convolutional Neural Network* (CNN). Dalam metode ini, sejumlah persentase tertentu dari aktivasi jaringan akan dinonaktifkan secara acak sehingga tidak ikut berkontribusi dalam proses pembelajaran. Teknik ini tidak diterapkan pada tahap pengujian (*testing*), di mana seluruh neuron kembali digunakan secara penuh. Penerapan *dropout* bertujuan untuk mencegah *overfitting* dengan cara mengurangi ketergantungan atau korelasi antar neuron dalam jaringan. Dengan demikian, model didorong untuk menangkap pola yang lebih luas dan tidak bergantung pada kombinasi neuron tertentu, sehingga mampu menghasilkan performa yang lebih baik ketika dihadapkan pada data baru [40].

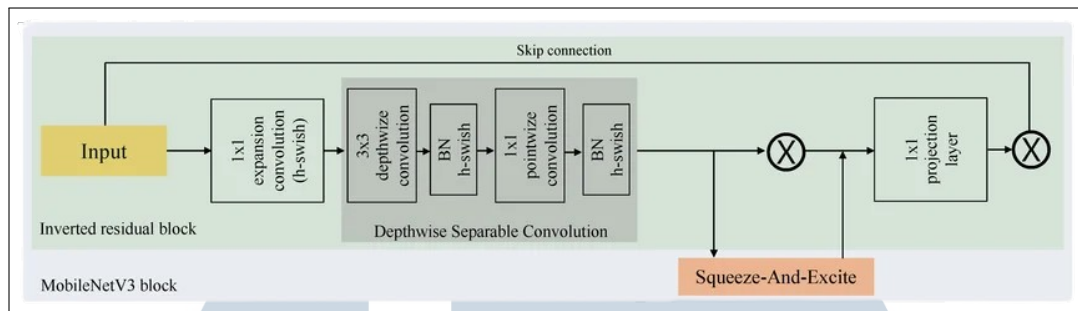
2.4 MobileNetV3-Small

MobileNetV3-Small adalah varian arsitektur *Convolutional Neural Network* (CNN) yang tergolong ringan dan dikembangkan melalui pendekatan *Neural Architecture Search* (NAS) serta teknik optimasi NetAdapt, dengan tujuan memperoleh hasil optimal antara ketepatan prediksi dan efisiensi komputasi. Pada tahap awal, arsitektur ini menggunakan *convolutional layer* pembuka yang berperan dalam *feature extraction* sebagai representasi fitur dasar dari citra masukan, mencakup informasi tepi maupun tekstur visual [41].

Proses ekstraksi fitur selanjutnya dilakukan melalui serangkaian *inverted residual bottleneck blocks* sebagai komponen utama arsitektur, yang secara sekuensial melibatkan tahap perluasan *channel*, *depthwise convolution* untuk pemrosesan fitur per-*channel* secara efisien, serta *projection layer* guna mengembalikan dimensi fitur ke ukuran yang lebih kompak. Mekanisme tersebut memungkinkan pengurangan kompleksitas komputasi secara signifikan tanpa mengorbankan kualitas representasi fitur yang dihasilkan [41].

Dalam upaya menekan jumlah operasi komputasi secara keseluruhan, MobileNetV3-Small menerapkan *depthwise separable convolution* sebagai strategi utama yang terbukti lebih efisien dibandingkan konvolusi standar. Arsitektur ini mengintegrasikan *Squeeze-and-Excitation* (SE), modul yang berfungsi melakukan pembobotan adaptif terhadap setiap *channel* fitur, sehingga representasi yang lebih relevan secara semantik dapat diperkuat selama proses pembelajaran berlangsung. *Activation function* hard-swish digunakan untuk menghadirkan non-linearitas yang lebih halus sekaligus meningkatkan stabilitas pelatihan model dibandingkan fungsi aktivasi konvensional [41].

Setelah tahap ekstraksi fitur selesai, *global average pooling* diterapkan guna mereduksi dimensi spasial *feature map* sebelum diteruskan ke tahap klasifikasi. Pada bagian akhir jaringan, fitur yang telah diringkas diproses oleh lapisan klasifikasi berupa *fully connected layer* atau *pointwise convolution* untuk menghasilkan keluaran prediksi kelas. Melalui kombinasi komponen-komponen tersebut, MobileNetV3-Small menjadi model yang ringan secara komputasi, efisien dalam penggunaan sumber daya, namun tetap efektif dalam mengekstraksi dan merepresentasikan fitur citra pada berbagai tugas pengenalan visual [41]. Arsitektur MobileNetV3-Small diperlihatkan di Gambar 2.10.



Gambar 2.10. Arsitektur MobileNetV3

Sumber: [42]

2.5 Hash MD5

Algoritma MD5 digunakan dalam proses data cleaning untuk mendeteksi dan menghapus data duplikat. Metode ini bekerja dengan cara menghasilkan nilai hash unik untuk setiap citra. Jika terdapat dua atau lebih citra yang memiliki nilai hash yang sama, maka citra tersebut dianggap identik atau duplikat, sehingga salah satunya dapat dihapus dari dataset. Dengan demikian, penggunaan MD5 membantu memastikan bahwa dataset hanya berisi data yang unik, sehingga dapat meningkatkan kualitas data dan kinerja model dalam proses pelatihan [43].

2.6 Augmentasi Data

Augmentasi data merupakan teknik dalam tahap preprocessing yang digunakan untuk meningkatkan variasi data dengan cara melakukan transformasi pada data asli tanpa mengubah makna atau informasi utamanya. Tujuan dari augmentasi data adalah untuk membantu model dalam mempelajari pola yang lebih beragam, meningkatkan kemampuan generalisasi, serta mengurangi risiko overfitting selama proses pelatihan [44].

1. *Horizontal flip*, yaitu memutar citra secara horizontal.
2. *Rotation*, yaitu melakukan rotasi pada citra
3. *Zoom*, yaitu memperbesar atau memperkecil citra.
4. *Shifting*, yaitu menggeser citra pada arah tinggi dan lebar.

2.7 Evaluasi Model

Evaluasi model merupakan salah satu tahap krusial dalam alur pengembangan sistem pembelajaran mesin yang digunakan untuk mengukur sejauh mana kemampuan model dalam menjalankan tugas klasifikasi. Proses evaluasi ini memanfaatkan sejumlah metrik yang bersumber dari *confusion matrix*, di antaranya *accuracy*, *precision*, *recall*, dan *F1-score* [45].

2.7.1 Confusion Matrix

Confusion matrix merupakan metrik evaluasi yang digunakan untuk mengukur kinerja model klasifikasi dengan cara membandingkan hasil prediksi model terhadap label aktual dari data uji. Selain format 2×2 yang hanya berlaku untuk permasalahan klasifikasi biner, *confusion matrix* juga dapat digunakan untuk permasalahan klasifikasi multikelas, seperti yang dicontohkan melalui kasus klasifikasi tiga kelas (Kelas A, Kelas B, dan Kelas C) pada Tabel 2.1.

Tabel 2.1. Confusion Matrix Multikelas (3×3)

		Actual		
		Class A	Class B	Class C
Predicted	Class A	AA	AB	AC
	Class B	BA	BB	BC
	Class C	CA	CB	CC

Sumber: [46]

Pada Tabel 2.1 merupakan representasi *confusion matrix* 3×3, baris menyatakan kelas hasil prediksi model, sedangkan kolom menyatakan kelas aktual dari data uji. Elemen-elemen yang terletak pada diagonal utama matriks menunjukkan jumlah data yang berhasil diklasifikasikan secara tepat oleh model untuk masing-masing kelas, sementara elemen-elemen di luar diagonal utama merepresentasikan jumlah kesalahan klasifikasi (*misclassification*), yakni data yang seharusnya termasuk ke dalam satu kelas namun oleh model justru diprediksi masuk ke kelas lainnya. Dengan demikian, semakin besar nilai pada diagonal utama dan semakin kecil nilai di luar diagonal, maka semakin baik pula performa model dalam melakukan klasifikasi [46].

2.7.2 Precision

Precision merupakan perbandingan antara banyaknya prediksi positif yang tepat sasaran dengan keseluruhan prediksi positif yang dikeluarkan oleh model. Metrik ini mencerminkan seberapa akurat model dalam menentukan suatu data sebagai bagian dari kelas positif [45]. Persamaan untuk menghitung nilai *precision* dapat dilihat pada 2.1.

$$Precision = \frac{TP}{TP + FP} \quad (2.1)$$

2.7.3 Recall

Recall merupakan perbandingan antara prediksi positif yang berhasil teridentifikasi dengan tepat terhadap keseluruhan data yang sesungguhnya berkelas positif, sehingga metrik ini mencerminkan sejauh mana kemampuan model dalam mengidentifikasi semua data yang tergolong positif [45]. Persamaan untuk menghitung nilai recall dapat dilihat pada 2.2.

$$Recall = \frac{TP}{TP + FN} \quad (2.2)$$

2.7.4 F1-Score

F1-score merupakan nilai rata-rata harmonik yang menggabungkan *precision* dan *recall*, yang berfungsi untuk menyeimbangkan kontribusi dari kedua metrik evaluasi tersebut, khususnya pada kondisi di mana distribusi kelas dalam data tidak merata. Semakin tinggi nilai *F1-score* yang diperoleh, mengindikasikan bahwa model mampu menghasilkan klasifikasi yang unggul baik dari sisi ketepatan prediksi maupun kemampuan cakupan kelasnya [45]. Rumus untuk menghitung F1-score disajikan pada 2.3.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (2.3)$$

2.7.5 Accuracy

Accuracy berfungsi untuk menghitung persentase prediksi yang berhasil diklasifikasikan dengan benar dari total keseluruhan data yang menjadi bahan pengujian. *Accuracy* menjadi salah satu metrik yang paling sering digunakan karena mampu merepresentasikan gambaran menyeluruh mengenai kinerja model secara global [45]. Rumus untuk menghitung nilai *accuracy* dapat dilihat pada 2.4.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.4)$$

