

BAB 2

LANDASAN TEORI

2.1 Sampah Anorganik

Sampah adalah hasil dari aktivitas manusia yang tidak bersifat biologis dan didefinisikan sebagai materi sisa yang tidak diinginkan setelah digunakan. Sampah anorganik berasal dari benda mati dan cenderung menjadi sampah kering yang sulit diuraikan (*nondegradable*)[3]. Jenis sampah ini memiliki potensi untuk didaur ulang dan memiliki nilai ekonomis, terutama sampah seperti kardus, logam, kertas, kaca, dan plastik. Sampah yang bersifat anorganik membutuhkan waktu yang lama untuk terurai dan volumenya terus meningkat setiap hari[6].

2.1.1 Sampah Kardus

Sampah kardus atau *cardboard* adalah limbah yang berasal dari bahan kardus atau karton, yang umumnya digunakan sebagai kemasan untuk berbagai produk. Sampah kardus biasanya berasal dari berbagai sumber seperti pengemasan produk konsumen, industry dan perdagangan, atau rumah tangga[6]. *Cardboard* ditunjukkan pada Gambar 2.1.



Gambar 2. 1 Sampah Kardus

2.1.2 Sampah Kaca

Kaca merupakan padatan yang berasal dari hasil olahan dari beberapa proses kimia, fisika, dan biologi. Bahan utama dari pembuatan kaca adalah pasir silika. Kaca memiliki efek positif dan efek negatif bagi lingkungan. Efek positif dari kaca adalah menunjang kehidupan masyarakat sedangkan efek negatif dari kaca adalah jika penggunaannya tidak sesuai maka akan berdampak serius pada lingkungan[6]. *Glass* ditunjukkan pada Gambar 2.2.



Gambar 2. 2 Sampah Kaca

2.1.3 Sampah Metal

Metal adalah bahan dasar yang berat, padat, memiliki sifat tertentu, dapat ditempa, dapat dilebur dengan menggunakan panas api dan listrik. Contoh dari bahan metal yang umum digunakan seperti stainless steel, aluminium, baja ringan, besi[6]. *Metal* ditunjukkan pada Gambar 2.3.



Gambar 2. 3 Sampah Metal

2.1.4 Sampah Kertas

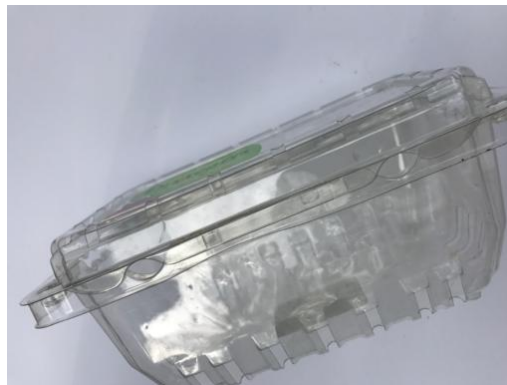
Kertas merupakan bahan yang umum sekali kita temui dengan bahan dasar pembuatannya dari kayu dan pohon. Penggunaan kertas yang meningkat menyebabkan meningkatnya kasus eksploitasi hutan[6]. *Paper* ditunjukkan pada Gambar 2.4.



Gambar 2. 4 Sampah Kertas

2.1.5 Sampah Plastik

Plastik salah satu material terbuat darinafta yang merupakan hasil dari turunan minyak bumi yang didapatkan dari proses penyulingan. Plastik termasuk material yang tidak alami karena tidak dapat digunakan kembali setelah digunakan dan sulit diuraikan oleh mikroba tanah sehingga dapat mencemari lingkungan[6]. *Plastic* ditunjukan pada Gambar 2.5.



Gambar 2. 5 Sampah Plastik

2.2 Citra Digital

Citra adalah representasi objek dua dimensi dari dunia visual, menyangkut berbagai macam disiplin ilmu yang mencakup seni, *human vision*, astronomi, teknik, dan sebagainya. Citra Merupakan suatu kumpulan piksel-piksel atau titik-titik yang berwarna yang berbentuk dua dimensi. Citra digital merupakan representasi numerik dari data citra agar dapat diolah, dimana citra digital dipresentasikan dengan menggunakan matriks $f(x,y)$ yang dimana x,y itu mempresentasikan sebuah piksel[9].

2.3 Image Processing

Image Processing merupakan proses yang digunakan untuk mengolah data citra asli sebelum diolah dengan menggunakan algoritma yang ada pada *Convolutional Neural Network*[10].

2.4 Resize

Resize dilakukan untuk mengubah ukuran citra dengan memperkecil ukuran citra pada arah horizontal dan/atau vertikal menjadi ukuran 224x224 piksel. Hal ini bertujuan untuk menyeragamkan ukuran dari masing-masing citra yang digunakan selama proses pelatihan dan pengujian dikarenakan terjadinya perbedaan ukuran akan mengakibatkan proses pelatihan yang lama dan tingkat akurasi akan mempengaruhi[11]. Pada penelitian ini, citra diubah menjadi ukuran 224x224 piksel sebagai input untuk arsitektur *DenseNet201*.

Untuk melakukan proses *resize image* menggunakan Rumus 2.1 dan 2.2 sebagai berikut.

$$S = \left(\frac{R}{Image} \right) \quad (2.1)$$

$$\begin{aligned} x' &= Sx \times x \\ y' &= Sy \times y \end{aligned} \quad (2.2)$$

Untuk mendapatkan hasil citra, pertama harus dicari terlebih dahulu nilai *S*. Di mana *R* merupakan target *resize* dan *Image* merupakan ukuran gambar yang akan dilakukan *resize*. Sedangkan *x'* melambangkan lebar *resize*, *y'* melambangkan panjang *resize*, *Sx* & *Sy* merupakan konstanta untuk mengubah ukuran, yang dimana *x* melambangkan lebar citra, dan *y* melambangkan panjang citra.

2.5 Augmentasi Citra

Augmentasi Citra merupakan proses pengolahan citra dengan teknik duplikasi dan modifikasi citra. Proses augmentasi data bertujuan untuk meningkatkan jumlah variasi citra sehingga memperluas dataset pelatihan[8].

2.6 Normalisasi Citra

Normalisasi pada citra merupakan proses untuk mengubah rentang nilai piksel dalam gambar menjadi rentang nilai antara 0 sampai 1, dilakukan normalisasi dengan membagi setiap nilai dengan angka 255 yang bertujuan untuk meringankan beban komputasi dalam tahap training. Sehingga mendapatkan nilai RGB pada rentang 0-1[12].

Berikut merupakan rumus normalisasi:

$$Normalize = \left(\frac{image}{255} \right) \quad (2.3)$$

2.7 Vertical Flip

Augmentasi *vertical flip* menghasilkan variasi dari citra asli di mana gambar dibalik secara vertikal sepanjang sumbu horizontal. Teknik ini digunakan sebagai bagian dari augmentasi data untuk memperbanyak variasi data pelatihan tanpa harus mengumpulkan gambar baru, yang dapat meningkatkan performa model machine learning, khususnya dalam tugas-tugas seperti pengenalan gambar, klasifikasi, dan deteksi objek[13].

$$y' = H - 1 - x \quad (2.4)$$

2.8 Horizontal Flip

Augmentasi *horizontal flip* menghasilkan variasi dari citra asli dengan memutar citra secara *horizontal (mirrored)* atau menghasilkan versi cermin dari citra tersebut. Secara matematis, untuk melakukan augmentasi *flip horizontal*, setiap piksel pada baris citra tetap berada di baris yang sama, tetapi posisi pikselnya[14].

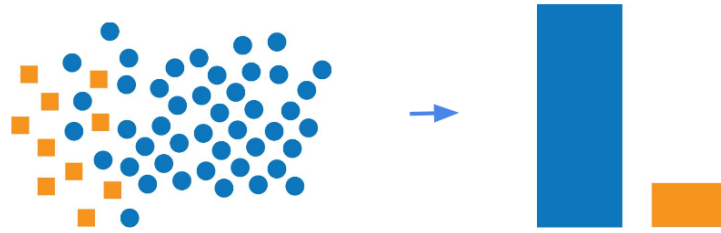
Berikut merupakan rumus horizontal *flip*:

$$x' = W - 1 - x \quad (2.5)$$

Dimana x' merupakan lebar hasil flip citra, y' merupakan panjang hasil rotasi citra, dan W merupakan lebar kolom citra.

2.9 Imbalanced Data

Imbalance data atau data yang tidak seimbang adalah kondisi di mana terdapat ketidakseimbangan jumlah antara kelas atau label dalam suatu dataset. Hal ini dapat menyebabkan masalah dalam pemodelan machine learning karena model cenderung lebih baik memprediksi kelas yang lebih banyak muncul dibandingkan dengan kelas yang lebih sedikit. Tugas-tugas pengolahan dan analisis data seperti klasifikasi, *clustering*, prediksi, dan lainnya menjadi tidak mudah apabila satu kelas memiliki jumlah data yang jauh lebih tinggi atau lebih rendah dibandingkan dengan kelas lain, sehingga dapat menyebabkan *imbalance data*[7].



Gambar 2. 6 Ilustrasi *Imbalanced Class*

(Sumber : Sachin D N, 2020)

2.10 *Class Weight*

Class weight adalah teknik yang dapat digunakan dalam pembelajaran mesin untuk mengatasi kelas yang tidak seimbang dengan cara memberikan bobot yang lebih tinggi pada sampel dari kelas minoritas. Tujuannya adalah untuk memberikan perhatian lebih pada kelas minoritas dengan memberikan bobot yang lebih tinggi, sehingga model lebih fokus dan sensitif terhadap kelas yang kurang representatif[7].

Berikut merupakan rumus *class weight*:

$$\left(\frac{\text{Total Sample}}{\text{Jumlah Kelas} * \text{Sample per Kelas}} \right) \quad (2.6)$$

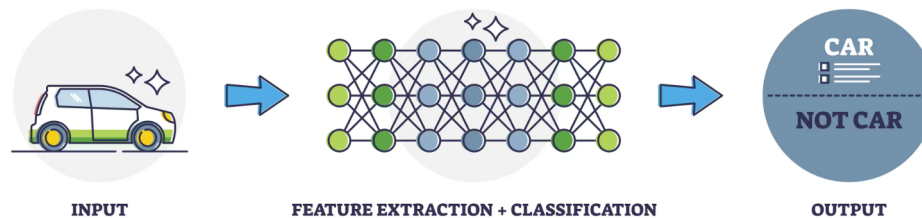
Rumus tersebut bertujuan untuk menghitung bobot kelas yang memperhitungkan jumlah sampel dalam kelas tersebut relatif terhadap total sampel dalam dataset dan jumlah kelas. Dengan menggunakan bobot ini, kelas dengan jumlah sampel yang lebih sedikit akan diberikan bobot yang lebih tinggi sehingga model dapat lebih fokus pada kelas-kelas minoritas.

2.11 *Deep Learning*

Deep learning adalah teknik dalam pembelajaran mesin yang menggunakan jaringan saraf tiruan dengan banyak lapisan untuk menghasilkan representasi yang semakin kompleks dari data. Dengan *deep learning*, mesin dapat secara efektif mempelajari fitur-fitur yang kompleks dan mendalam dari jumlah data yang sangat besar secara otomatis[15].

Deep Learning menggunakan algoritma *backpropagation* menunjukkan suatu mesin harus mengubah parameter yang digunakan untuk menghitung nilai di setiap lapisan dari nilai di lapisan sebelumnya atau proses nya berjalan mundur pada setiap lapisannya. *Deep learning* terbagi menjadi tiga kategori pendekatan yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*[16]. Salah satunya potensi dari *deep learning* adalah

mengganti fitur buatan tangan dengan algoritma yang efisien untuk pembelajaran hierarkis *unsupervised* (fitur tanpa pengawasan) atau *semi-supervised feature learning* (semi diawasi) dan *hierarchical feature extraction* (ekstraksi fitur). Penerapan deep learning penerapan deep learning telah digunakan dalam beberapa bidang seperti klasifikasi gambar, klasifikasi video, object detection, object recognition, text-to-speech, natural language processing, robotic, text classification, dan singing synthesis[17].

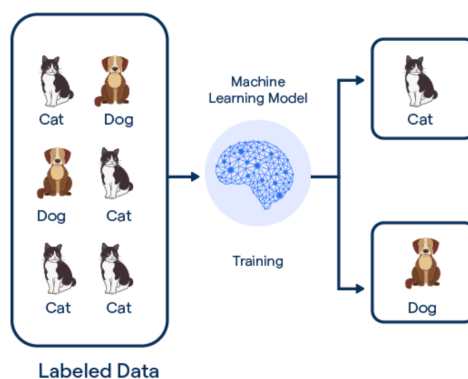


Gambar 2. 7 Deep Learning

(Sumber : Diego Gasco, 2022)

2.12 Supervised Learning

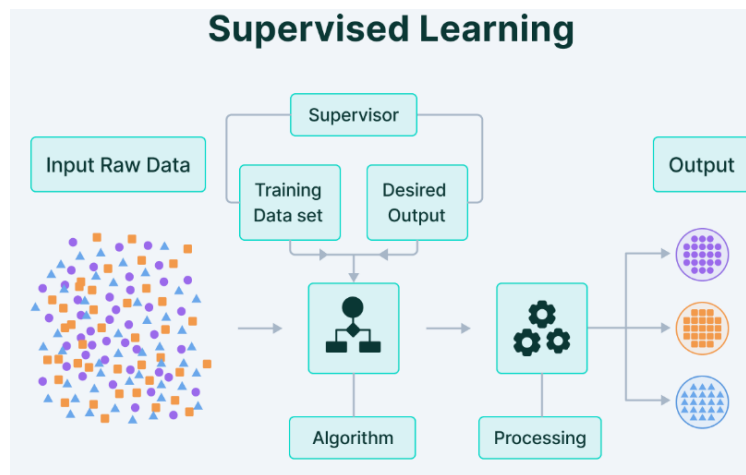
Supervised Learning merupakan algoritma yang membangkitkan suatu fungsi yang memetakan input ke output yang diinginkan. Supervised learning adalah salah satu jenis pembelajaran mesin (*machine learning*) di mana model dilatih menggunakan data yang sudah diberi label. Setiap contoh dalam data pelatihan memiliki input dan output yang diinginkan. Tujuan dari *supervised learning* adalah untuk membuat model yang dapat memprediksi output yang benar untuk input baru berdasarkan pola yang telah dipelajari dari data pelatihan[18].



Gambar 2. 8 Supervised Learning

(Sumber : Nikita, 2023)

Cara kerja *supervised learning* di mana model pembelajaran mesin belajar dari data yang sudah diberi label untuk membuat prediksi pada data baru. Selama pelatihan, model memproses setiap input dan menghasilkan prediksi, kemudian membandingkannya dengan output yang sebenarnya. Perbedaan antara prediksi dan output ini diukur menggunakan fungsi loss, yang menunjukkan seberapa baik atau buruk prediksi model. Menggunakan algoritma optimisasi, untuk menyesuaikan bobot dan parameter internalnya dengan tujuan meminimalkan fungsi loss. Proses ini berulang secara iteratif untuk seluruh dataset pelatihan, sehingga model terus belajar dan memperbaiki prediksinya[18].

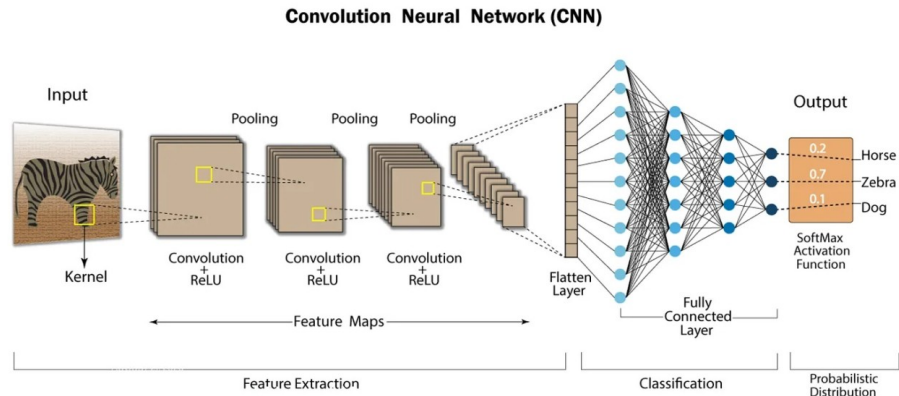


Gambar 2. 9 Alur Proses *Supervised Learning*

(Sumber : Yogesh Sharma, 2023)

2.13 *Convolutional Neural Network*

Convolution Neural Network merupakan suatu pengembangan dari sebuah *Multilayer Perceptron (MLP)* yang diciptakan atau didesain untuk mengolah data dua dimensi. Metode CNN ini banyak digunakan dalam suatu pengolahan citra. CNN merupakan suatu algoritma yang berasal dari *Deep Learning* dan menghasilkan suatu data yang signifikan dalam pengenalan suatu citra. Tentunya hal tersebut dikarenakan metode ini menerapkan sistem yang sama dengan pengenalan citra pada *visual cortex* yang ada pada manusia. Tentunya hal tersebut menyebabkan metode ini memiliki kemampuan yang hampir sama dengan cara kerja manusia dan juga tentunya dengan hasil yang lebih baik dan lebih konsisten[6].



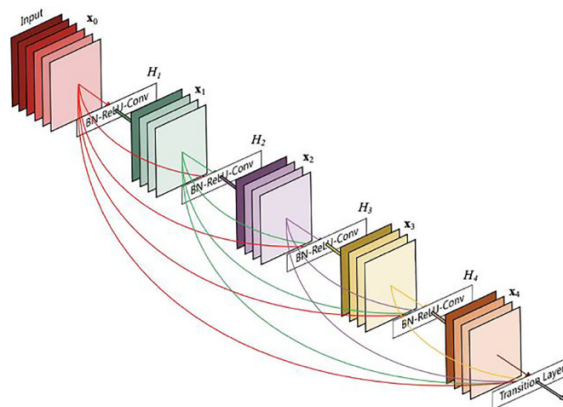
Gambar 2. 10 Convolutional Neural Network

Sumber: (Nafizul Haque, 2023)

Pada gambar 2.9 terlihat secara garis besar terdapat dua proses pada CNN. Proses pertama adalah *Feature Extraction* yang meliputi *convolutional* dan *pooling*. Kemudian proses kedua yaitu *Classification* yang meliputi *flatten*, dan *fully connected layer*.

2.14 DenseNet201

Densely Convolutional Network merupakan arsitektur *Deep Learning* yang menghubungkan setiap layer beserta *feature-maps* ke seluruh layer berikutnya. Layer berikutnya akan menerima input *feature-maps* dari seluruh layer sebelumnya. DenseNet memiliki beberapa keunggulan menarik yaitu meringankan masalah gradien-gradien, memperkuat penyebaran fitur, mendorong penggunaan kembali fitur, dan secara substansial mengurangi jumlah parameter[8]

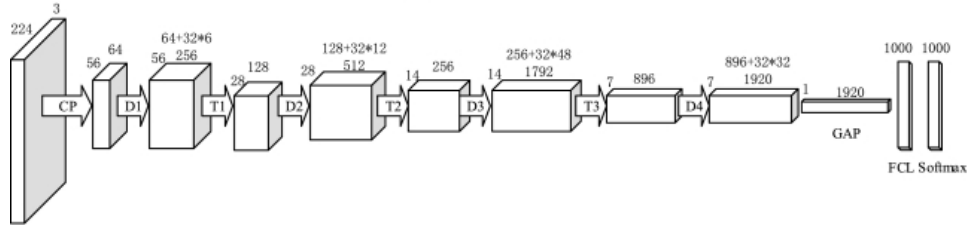


Gambar 2. 11 Arsitektur DenseNet

Sumber: (Sarra Rouabhi, 2023)

Pada gambar 2.11 untuk setiap komposisi lapisan menggunakan *batch normalization*, *ReLU activation* dan *convolution* dengan filter 3x3. Pada setiap blok ada masukan berupa

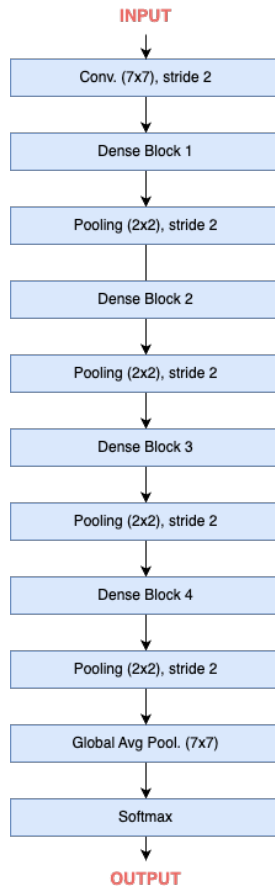
matriks sesuai dengan pixel citra kemudian masuk ke proses *batch normalization* untuk mengurangi adanya *overfitting* pada saat proses *training*, *ReLU activation* digunakan untuk mengubah nilai x menjadi 0 jika nilai x tersebut bernilai negatif, sedangkan sebaliknya untuk nilai x tetap dipertahankan apabila nilai tidak kurang dari 0[11].



Gambar 2. 12 DenseNet201

Sumber: (Radhika Natarajan, 2022)

Gambar 2.10 menunjukkan bahwa DenseNet201 merupakan $5 + (6 + 12 + 48 + 32) * 2 = 201$, dengan penjelasan bahwa 5 merupakan Konvolusi dan *Pooling Layer*. 3 merupakan lapisan transisi(6, 12, 48). 1 merupakan *Classification Layer*(32). 2 merupakan *DenseBlock* (Konvolusi 1x1 dan 3x3). Perbedaan signifikan DenseNet201 dengan DenseNet variasi yang lain terletak pada jumlah parameter dan kedalaman model. DenseNet201 memiliki lebih banyak parameter dibandingkan dengan DenseNet121 atau DenseNet169, ini dikarenakan oleh jumlah blok dan lapisan yang dimiliki oleh DenseNet201 yang lebih banyak dari pendahulunya dan DenseNet201 memiliki lebih banyak lapisan daripada DenseNet121 atau DenseNet169, kedalaman model dapat mempengaruhi kapasitas model dan kemampuannya untuk mengekstraksi fitur yang kompleks[11]. Gambar 2.11 merupakan arsitektur DenseNet201.



Gambar 2. 13 Arsitektur DenseNet201

2.15 Convolutional Layer

Convolutional Layer ini merupakan suatu operasi konvolusi suatu output dari lapisan sebelumnya. *Layer* ini merupakan suatu proses utama yang mendasari suatu CNN dan merupakan lapisan utama yang paling penting dalam suatu sistem yang digunakan dalam CNN. Proses konvolusi adalah proses yang mengaplikasikan filter pada gambar. Pada proses konvolusi ada perkalian matriks terhadap filter dan area pada gambar[6]. *Convolution Matrix* ditunjukkan pada gambar 2.14.

23	25	0	21	22
23	25	0	0	25
25	25	0	20	20
20	25	24	0	25
25	24	0	25	24

 $\ast$

1	1	1
-1	1	-1
1	-1	1

 $=$

125	65	38
117	29	74
80	93	40

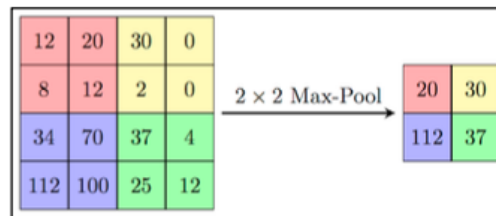
Gambar 2. 14 Ilustrasi *Convolutional Layer*

Sumber: (Budi Yanto, 2021)

2.16 Pooling Layer

Pooling merupakan metode yang bertujuan mengurangi ukuran *feature map* dari hasil proses konvolusi. Dalam *pooling* terdapat dua metode yaitu *average pooling* dan *max pooling*. *Average pooling* mengambil nilai rata-rata sedangkan *max pooling* mengambil nilai maksimal[19]. Pada gambar 2.11 diilustrasikan proses dari *max pooling*.

Pooling adalah untuk mencapai invarian spasial dengan mengurangi resolusi peta fitur. Setiap peta fitur yang dikumpulkan sesuai dengan satu peta fitur dari lapisan sebelumnya. Dalam sebagian besar CNN, metode subsampling yang digunakan adalah *max pooling*. *Max pooling* membagi output dari *convolution* layer menjadi beberapa *grid* kecil lalu mengambil nilai maksimal dari setiap *grid* untuk menyusun matriks citra yang telah direduksi. Sedangkan *average pooling* membagi output menjadi beberapa *grid* kecil untuk mengambil nilai rata-rata dari setiap *grid* yang ada pada citra untuk menyusun matriks yang sudah direduksi. Pada Gambar 2.15 diilustrasikan proses dari *max pooling*.



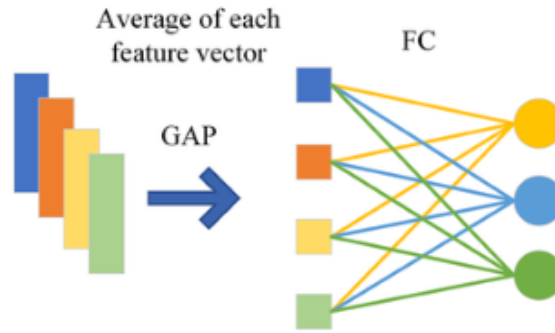
Gambar 2. 15 Ilustrasi Max Pooling

Sumber: (Elizabeth Endri, 2020)

2.17 Global Average Pooling

Lapisan *global average pooling* adalah *layer* pada CNN yang akan menghasilkan satu *feature map* untuk setiap kategori tugas klasifikasi yang ada pada lapisan terakhir dengan cara mengambil nilai rata-rata dari setiap *feature map* lalu hasil vektornya akan diberikan ke lapisan *fully connected layer* lalu menuju *softmax*. *Global average pooling* dirancang untuk mengurangi dimensi data dari fitur map terakhir dalam jaringan konvolusional dengan menghitung rata-rata dari seluruh elemen dalam setiap fitur map. Dengan menghitung rata-rata dari setiap fitur map, *global average pooling* membantu mengurangi *overfitting*, mengurangi jumlah parameter, dan menjaga informasi spasial dari data input[20].

Berikut Gambar 2.16 merupakan gambar proses *global average pooling* menuju *Fully Connected*.



Gambar 2. 16 Ilustrasi Global Average Pooling

Sumber: (Jingkang Liang, 2022)

2.18 Batch Normalization

Lapisan normalisasi *batch* atau *batch normalization* digunakan untuk membuat pelatihan jaringan syaraf tiruan menjadi lebih cepat dan stabil melalui normalisasi input dari setiap lapisan, dengan melakukan pemusatan dan penskalaan ulang. Lapisan tersebut mempercepat proses pelatihan di dalam jaringan dan membantu untuk mendapatkan stabilitas. Normalisasi *batch* bertujuan untuk mengurangi pergeseran kovariat internal dengan mencoba menjaga distribusi output dari lapisan sebelumnya tidak berubah[21].

2.19 Rectified Linear Unit (ReLU)

Rectification Linear Unit (ReLU) merupakan sebuah fungsi yang bertujuan mengenalkan non-linearitas dan meningkatkan representasi dari model. *ReLU activation* digunakan untuk mengubah nilai x menjadi 0 jika nilai x tersebut bernilai negatif, sedangkan sebaliknya untuk nilai x tetap dipertahankan apabila nilai tidak kurang dari 0[19].

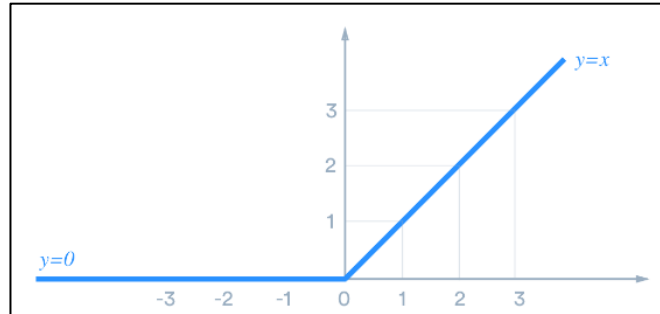
Berikut adalah persamaan ReLU:

$$f(x) = \max(0, x) \quad (2.7)$$

Keterangan:

$f(x)$ = nilai dari ReLU activation

x = nilai dari matriks citra



Gambar 2. 17 Fungsi Aktivasi ReLU

Sumber: (Oona Voican, 2021)

2.20 Softmax Activation

Softmax diterapkan pada lapisan terakhir pada jaringan saraf. *Softmax* lebih dari itu umum digunakan daripada *ReLU*, *sigmoid* atau *tanh*. Ini digunakan untuk menghitung distribusi probabilitas dari *vector* bilangan real. Fungsi *softmax* menghasilkan output yang merupakan kisaran nilai antara 0 dan 1, dengan jumlah probabilitas sama dengan 1[11].

Berikut merupakan rumus softmax:

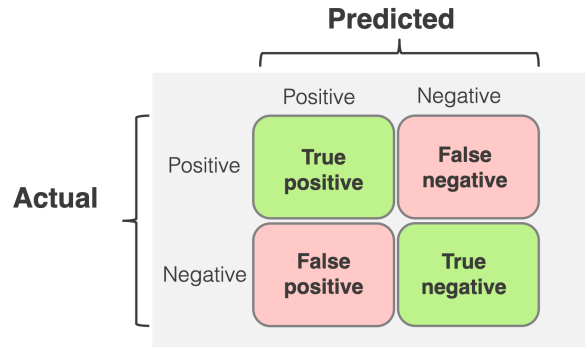
$$s(x_n) = \frac{e^{\text{avgpool}-n}}{\text{denominator}} \quad (2.8)$$

2.21 Evaluasi Kinerja Model

Evaluasi kinerja model merupakan yang terpenting dalam menentukan suatu model bagus atau tidak. Pada kasus klasifikasi, evaluasi kinerja yang digunakan berupa *Confusion matrix*, *precision*, *recall*, *accuracy* dan *F1-score*[19]. Terdapat beberapa pengukuran kinerja model yang digunakan dalam penelitian ini.

2.21.1 Confusion Matrix

Proses ini akan dilakukan pada saat *training* untuk mengukur performa dari model yang selesai *ditrain*. *Confusion matrix* digunakan untuk menghitung *accuracy*, *recall*, *precision* dan juga *F1-Score*. *Confusion matrix* yang ditampilkan berdasarkan model yang sudah *ditrain* lalu akan memprediksi data *test* yang sudah disediakan bersama labelnya[22].



Gambar 2. 18 Ilustrasi *Confusion Matrix*

Sumber: (Handika Sanjaya, 2021)

Terdapat beberapa istilah umum yang digunakan, yaitu:

- True Positive (TP) – Data positif yang diprediksi positif.
- True Negative (TN) - Data negatif yang diprediksi negatif.
- False Positive (FP) - Data negatif namun diprediksi sebagai data positif.
- False Negative (FN) - data positif namun diprediksi sebagai data negatif.

2.21.2 Accuracy

Accuracy merupakan rasio prediksi benar (positif dan negatif) dari keseluruhan data. *Accuracy* mewakili jumlah *instance* data yang diklasifikasi dengan benar diatas jumlah total *instance* data[19]. Berikut rumus 2.9 merupakan rumus *Accuracy*:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (2.9)$$

2.21.3 Precision

Precision merupakan suatu informasi yang diambil pada salah satu bagian data yang dimana rasio prediksi benar positif (TP) pada kelas dibandingkan dengan keseluruhan hasil yang di prediksi positif kelas. Berikut merupakan rumus *precision*[19]. Berikut rumus 2.10 merupakan rumus *precision*.

$$Precision = \left(\frac{TP}{TP + FP} \right) \times 100\% \quad (2.10)$$

2.21.4 Recall

Recall merupakan seberapa akurat kinerja suatu model untuk mengklasifikasi benar atau menghitung banyaknya kekeliruan klasifikasi *false negative* pada model. Perhitungan *recall* dapat dilihat pada persamaan dibawah ini[19]. Berikut Rumus 2.11 merupakan rumus *recall*.

$$Recall = \left(\frac{TP}{TP + FN} \right) \times 100\% \quad (2.11)$$

2.21.5 F-1 Score

F1-Score merupakan metrik gabungan yang memadukan *precision* dan *recall* menjadi satu skor tunggal. Berguna untuk mencari keseimbangan antara ketepatan (*precision*) dan kelengkapan (*recall*) dalam model. *F1-score* adalah *harmonic mean* (rata-rata harmonik) dari *precision* dan *recall*, ini membuatnya lebih informatif daripada *accuracy* ketika berhadapan dengan kelas yang tidak seimbang, karena menggabungkan kedua metrik penting ini ke dalam satu angka[19]. Berikut rumus 2.12 merupakan rumus *f1-score*.

$$F - 1 \text{ score} = \frac{2 \times (Precision \times recall)}{Precision + Recall} \times 100\% \quad (2.12)$$