

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1. Buah Apel**

Apel merupakan salah satu jenis buah yang kandungan lemak jenuhnya rendah dan nol kolesterol serta memiliki berbagai manfaat dalam menunjang kesehatan diantaranya meningkatkan kesehatan jantung, mencegah asma, membersihkan hati, mengurangi kolesterol, serta membantu imunitas tubuh manusia.[4] Umumnya kondisi ideal untuk mengetahui apel segar yaitu dengan melihat permukaan kulit apel yang mulus tanpa ada noda maupun sayatan dan lubang, maka dapat dikatakan apel tersebut dalam keadaan segar dan memiliki kualitas baik untuk dapat dikonsumsi. Buah apel haruslah memiliki bentuk dan kualitas yang baik sehingga aman dan lezat untuk dikonsumsi.

#### **2.2. Kualitas Buah Apel**

Buah apel yang berkualitas dapat dikenali dari berbagai aspek fisiknya. Pertama, kulit apel yang halus, bersih, dan berkilau menandakan kesegarannya. Selain itu, tekstur kulit yang kencang mengindikasikan kematangan yang baik. Ukuran dan bentuk apel yang konsisten juga menunjukkan kualitas yang baik, meskipun ukuran dapat bervariasi tergantung pada jenisnya. Buah apel yang berkualitas biasanya terasa berat dan padat saat dipegang. Pastikan juga apel bebas dari cacat fisik seperti bintik-bintik busuk, goresan, atau keriput yang signifikan. Hal yang harus diperhatikan tangkai apel; tangkai yang kuat menunjukkan bahwa apel ini dipetik pada waktu yang tepat. Memperhatikan aspek fisik ini saat memilih apel akan membantu Anda mendapatkan buah yang segar, matang, dan enak untuk dinikmati.

#### **2.3. Kulit Buah Apel**

Kulit buah apel adalah bagian luar yang melindungi dan menyelubungi daging buah di dalamnya. Struktur kulit apel terdiri dari lapisan epidermis, yang merupakan lapisan terluar yang terdiri dari sel-sel yang rapat dan kuat. Kulit apel memiliki beberapa fitur fisik yang mencerminkan kualitas dan kesehatan buah. Warna kulit apel bervariasi tergantung pada

jenisnya, mulai dari merah, hijau, kuning, hingga kombinasi warna-warna ini. Warna pada kulit apel disebabkan oleh pigmen seperti antosianin (yang memberikan warna merah atau ungu) dan klorofil (yang memberikan warna hijau). Kandungan pigmen ini dapat berubah seiring dengan kematangan buah.

### **2.3.1. Kulit Buah Apel Kualitas Baik**

Apel yang berkualitas baik memiliki warna kulit yang cerah dan merata. Warna apel bisa bervariasi tergantung pada jenisnya, tetapi apel yang matang umumnya memiliki warna yang intens dan menarik, tanpa bintik-bintik gelap atau area yang terlalu pucat.

### **2.3.2. Kulit Buah Apel Kualitas Buruk**

Kulit apel dengan kualitas buruk dapat dikenali melalui beberapa ciri khas seperti bercak cokelat atau hitam yang menunjukkan adanya penyakit atau kerusakan fisik, kulit yang keriput atau kusut menandakan apel kehilangan kelembapan, serta bercak lembek yang mengindikasikan pembusukan. Perubahan warna menjadi kusam, kondisi kulit yang kasar atau pecah-pecah, dan adanya jamur atau cendawan juga menandakan kualitas yang menurun. Selain itu, bau tidak sedap yang berasal dari apel merupakan indikasi bahwa buah tersebut sudah membusuk. Semua tanda ini menunjukkan bahwa bagian dalam apel mungkin sudah tidak layak dikonsumsi.

## **2.4. Citra Digital**

Citra digital adalah representasi gambar yang dibentuk melalui proses sampling dan kuantisasi. Sampling mengacu pada pengumpulan data gambar dalam bentuk matriks yang terdiri dari baris dan kolom. Dalam konteks gambar digital, sampling mencakup pembentukan piksel (titik-titik) pada gambar. Sementara itu, kuantisasi mengacu pada pengonversian nilai kecerahan gambar ke dalam skala abu-abu menggunakan digit biner. Kuantisasi juga dapat menggambarkan pengelompokan warna dalam gambar berdasarkan jumlah nilai yang direpresentasikan.[8]

## **2.5. Pre Processing Data**

Data preprocessing adalah proses konversi data mentah ke dalam format yang lebih mudah dipahami dan dapat digunakan untuk analisis lebih lanjut. Tujuan utama dari proses ini adalah untuk mengatasi kesalahan umum yang sering terjadi pada data mentah, seperti kekurangan

informasi, format yang tidak teratur, atau ketidaksesuaian dengan format yang dibutuhkan untuk analisis atau pengolahan data selanjutnya.[14]

### 2.5.1. *Resize*

Resize dilakukan pada penelitian ini untuk perubahan ukuran pada keseluruhan dataset gambar, menjadi dimensi 224x224 pixel dari jumlah total 2,275 dataset. Gambar yang telah mengalami perubahan ukuran tersebut kemudian disimpan dalam dataset baru yang diberi nama "Resized Image". [15] Untuk melakukan proses resize image menggunakan Rumus 2.1 dan 2.2 sebagai berikut.

$$x' = Sx \times x \quad (2.1)$$

$$y' = Sy \times y \quad (2.2)$$

Dimana  $x'$  melambangkan lebar resize,  $y'$  melambangkan panjang hasil resize,  $Sx$  &  $Sy$  merupakan konstanta untuk mengubah ukuran, yang dimana  $x$  melambangkan lebar citra, dan  $y$  melambangkan panjang citra.

## 2.7. *Augmentasi Data*

Salah satu strategi untuk meningkatkan kinerja dan kapabilitas model klasifikasi adalah dengan menambah jumlah data pelatihan. *Image augmentation* dikenal efektif terutama dalam proses klasifikasi yang menggunakan metode CNN dan memiliki dataset kurang dari 1 juta[16]. Dalam penelitian ini, jumlah total gambar dalam dataset untuk setiap klasifikasi adalah sebanyak 1.100 gambar kurang lebih, sehingga pilihan image augmentation diadopsi dengan maksud untuk memperluas dataset pelatihan, yang diharapkan dapat meningkatkan akurasi model. Image augmentation memiliki kemampuan untuk secara otomatis menghasilkan dataset pelatihan baru dari dataset yang telah ada sebelumnya. Proses augmentasi gambar melibatkan serangkaian operasi dalam manipulasi gambar, seperti pergeseran, pembalikan, *zoom*, dan variasi lainnya. Dalam penelitian ini menggunakan beberapa transformasi yaitu *rescale*, *shear range*, *zoom range*, *horizontal flip*, dan *vertical flip*.[14]

### 2.7.1. *Rescale*

Rescale pada citra adalah proses mengubah rentang nilai piksel gambar ke rentang yang lebih kecil atau lebih besar, biasanya antara 0 dan 1. Teknik ini dilakukan dengan membagi setiap

nilai piksel dengan angka 255. Tujuannya adalah untuk meringankan beban komputasi selama tahap pelatihan model[13].

Berikut rumus untuk *rescale*:

$$Rescale = image \times \left( \frac{1.0}{225} \right) \quad (2.3)$$

### 2.7.2. Zoom Range

Metode Zoom Range digunakan untuk memperbesar gambar dengan menambah beberapa pixel disekitar gambar. Augmentasi zoom range mengontrol seberapa besar citra dapat diperbesar (zoom in) atau diperkecil (zoom out) selama proses augmentasi. Dalam konteks augmentasi data, zoom range mengacu pada faktor skala yang diterapkan pada gambar. Zooming dapat dilakukan dengan memperbesar atau memperkecil gambar, dan biasanya dinyatakan sebagai rentang antara dua nilai (misalnya, 0.8 hingga 1.2). Zoom in (perbesar) berarti nilai faktor zoom lebih dari 1 (misalnya, 1.2 berarti memperbesar gambar sebesar 20%). Zoom out (perkecil) yang artinya nilai faktor zoom kurang dari 1 (misalnya, 0.8 berarti memperkecil gambar sebesar 20%).[13].

Untuk menghitung koordinat baru dari piksel setelah zoom, kita perlu menentukan faktor zoom (zoom factor). Misalkan  $Z$  adalah faktor zoom, maka:

- Jika  $Z > 1$ , gambar diperbesar.
- Jika  $Z < 1$ , gambar diperkecil.

Koordinat baru ( $x'$ ,  $y'$ ) dari piksel ( $x$ ,  $y$ ) pada gambar yang di-zoom dapat dihitung dengan:

$$x' = \frac{x - center\_x}{Z} + center\_x$$

$$y' = \frac{y - center\_y}{Z} + center\_y$$

Di mana  $center\_x$  dan  $center\_y$  adalah koordinat pusat dari gambar asli.

### 2.7.3. Horizontal flip

Augmentasi flip horizontal menghasilkan variasi citra dengan memutar gambar secara horizontal, menciptakan efek cermin dari gambar asli. Secara matematis, augmentasi ini dilakukan dengan mempertahankan setiap piksel pada baris yang sama, tetapi membalik urutan posisi piksel di dalam baris tersebut. Misalnya, piksel pertama menjadi piksel terakhir, piksel kedua menjadi piksel kedua terakhir, dan seterusnya.[13]

Berikut adalah rumus dari *horizontal flip* :

$$x' = w - 1 - x \quad (2.4)$$

$$y' = y(\text{nilai koordinat } y \text{ tetap}) \quad (2.5)$$

### 2.7.4. Vertikal Flip

Augmentasi vertikal flip adalah salah satu teknik augmentasi yang sederhana namun efektif. Teknik ini melibatkan pembalikan citra secara vertikal, menghasilkan gambar yang merupakan cerminan dari gambar aslinya sepanjang sumbu horizontal. Vertikal flip membalik citra di sepanjang sumbu horizontal, artinya piksel di bagian atas citra dipertukarkan dengan piksel di bagian bawah citra. Misalnya, baris pertama citra menjadi baris terakhir, baris kedua menjadi baris kedua terakhir, dan seterusnya. [13]

Secara matematis, jika kita memiliki citra dengan ukuran  $H \times W$  (Tinggi  $H$  dan lebar  $W$ ), maka koordinat piksel  $(x,y)$  pada citra asli akan dipindahkan ke koordinat  $(H-1-x,y)$  pada citra hasil flip vertikal. Berikut adalah rumusnya:

$$\text{Flip\_Vertikal}(x, y) = (H - 1 - x, y) \quad (2.5)$$

### 2.7.5. Shear Range

*Shear range* dalam augmentasi data gambar adalah transformasi geometris yang menggeser lapisan gambar seolah-olah gambar tersebut diregangkan atau dimiringkan ke satu arah tertentu. Ini menciptakan efek miring pada citra asli dengan menyelaraskan baris atau kolom secara proporsional, yang dapat membantu model belajar mengenali objek dalam berbagai orientasi[13]. Berikut adalah rumus untuk *shear range* :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & s_x \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + s_x \cdot y \\ y \end{bmatrix} \quad (2.6)$$

## 2.8. Deep Learning

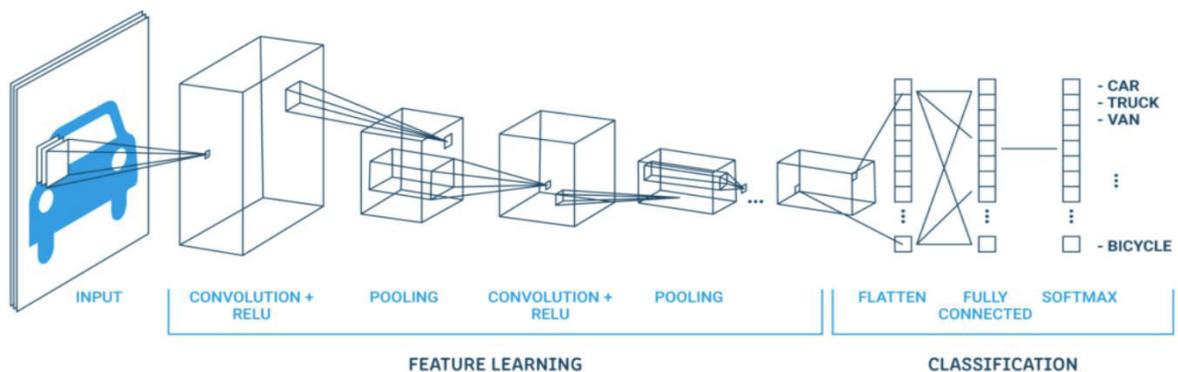
Deep Learning merupakan jenis metode pembelajaran mesin yang berfokus pada representasi data pembelajaran. Konsep Deep Learning menggunakan jaringan saraf tiruan (artificial neural networks), terutama jaringan saraf konvolusional mendalam (CNN), telah membuktikan kesuksesannya dalam beberapa tahun terakhir di berbagai aplikasi visi komputer, seperti klasifikasi gambar, deteksi objek, dan segmentasi semantik. Algoritma Deep Learning, terutama CNN, telah berhasil melewati banyak evaluasi benchmark terkenal, contohnya adalah ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Ini menandai perbedaan dengan sistem visi komputer tradisional yang lebih bergantung pada desain manual. Deep Learning memberikan kemampuan untuk belajar secara otomatis dari data yang diberikan, memungkinkan sistem untuk mengidentifikasi pola yang rumit dan fitur yang penting secara hierarkis. Penerapan Deep Learning, terutama dalam konteks CNN, telah membuka pintu untuk kemajuan besar dalam bidang visi komputer dan aplikasi lainnya, dengan menghadirkan tingkat akurasi dan kinerja yang signifikan dalam pengolahan gambar dan pengenalan pola. [17].

## 2.9. Convolutional Neural Network

Convolutional Neural Network (CNN) adalah pengembangan dari Multilayer Perceptron (MLP) yang dirancang khusus untuk mengolah data dua dimensi, seperti citra. CNN termasuk dalam jenis Deep Neural Network karena memiliki kedalaman jaringan yang tinggi dan sering digunakan dalam pemrosesan data citra. Penggunaan CNN dalam klasifikasi citra lebih menguntungkan daripada MLP karena CNN mampu menyimpan informasi spasial dari data citra. Sebaliknya, MLP menganggap setiap piksel sebagai fitur yang independen dan tidak memperhitungkan korelasi spasial antara piksel-piksel dalam citra, yang dapat menghasilkan performa klasifikasi yang kurang optimal.[2] CNN menggunakan operasi konvolusi untuk mengekstraksi fitur-fitur lokal dari citra secara efisien, sehingga mampu menangkap informasi

spasial yang penting untuk klasifikasi yang lebih akurat. Dengan demikian, kelebihan utama CNN dibandingkan dengan MLP dalam konteks pengolahan citra adalah kemampuannya untuk mempertahankan informasi spasial, mengakibatkan hasil klasifikasi yang lebih baik dan lebih tepat dalam tugas-tugas pengenalan dan klasifikasi citra.

CNN terdiri dari beberapa lapisan yang berturut-turut melakukan operasi konvolusi, aktivasi non-linear, dan pooling. Lapisan konvolusi adalah inti dari CNN dan berfungsi untuk mengekstraksi fitur-fitur penting dari data input. Setiap lapisan konvolusi terdiri dari sejumlah filter (kernel) yang diterapkan pada gambar secara bergeser untuk menghasilkan peta fitur[17]. Gambar 1.2 menunjukkan contoh dari sebuah jaringan saraf tiruan.



**Gambar 2.1. Ilustrasi Arsitektur CNN**

Secara garis besar, Convolutional Neural Network (CNN) memiliki dua proses utama. Proses pertama adalah Feature Extraction yang melibatkan konvolusi (convolutional layer) untuk mengekstraksi fitur-fitur penting dari data seperti gambar, serta pooling untuk mengurangi dimensi spasial dari peta fitur. Proses kedua adalah Classification, di mana fitur-fitur yang telah diekstraksi diubah menjadi vektor satu dimensi (flatten layer) dan kemudian diproses melalui lapisan-lapisan fully connected (dense) untuk melakukan klasifikasi atau prediksi terhadap data input. Dengan demikian, CNN dapat secara efektif memahami representasi-fitur dari data input dan menghasilkan output yang relevan sesuai dengan tugas yang diberikan, seperti klasifikasi gambar.

### 2.9.1. Convolutional Layer

Proses Convolution layer adalah komponen kunci dalam Convolutional Neural Network (CNN) yang terdiri dari sejumlah filter yang diinisialisasi secara acak. Filter ini melakukan operasi konvolusi pada input gambar dengan tujuan ekstraksi fitur untuk mempelajari representasi-fitur dari gambar tersebut. Di dalam convolution layer, neuron-neuron terorganisir menjadi peta fitur (feature maps). Setiap neuron pada peta fitur ini memiliki receptive field yang terhubung dengan neuron-neuron pada convolution layer sebelumnya melalui serangkaian bobot yang ditentukan secara terlatih. Serangkaian bobot ini juga dikenal sebagai filter bank gambar.[18] Persamaan matematis yang mengilustrasikan operasi konvolusi pada convolution layer seperti pada persamaan berikut :

$$h(x, y) = f(x, y) * g(x, y) \quad (2.7)$$

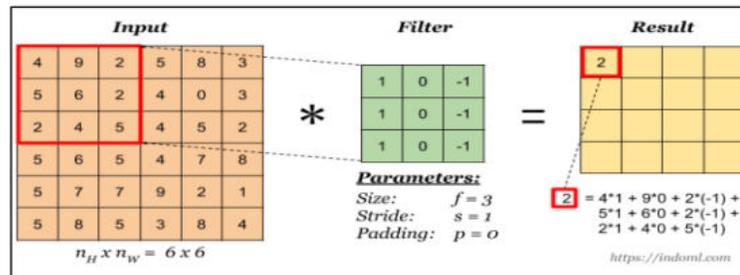
Di mana:

$h(x,y)$  = hasil dari proses convolution

$f(x,y)$ = nilai matrix citra

$g(x,y)$ = kernel convolution

Proses konvolusi melibatkan menggeser filter secara bertahap (disebut juga sebagai kernel) pada seluruh peta fitur input dan melakukan perkalian titik antara filter dan bagian yang sedang diproses dari input. Hasil perkalian titik ini dijumlahkan dan ditambahkan dengan bias, lalu diterapkan fungsi aktivasi untuk menghasilkan output yang akhirnya membentuk peta fitur.



Gambar 2.8. Ilustrasi Operasi Convolution

### 2.9.2. Pooling

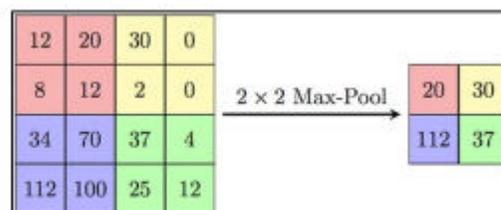
Pooling digunakan dalam Convolutional Neural Network (CNN) untuk mencapai invarian spasial dengan mengurangi resolusi peta fitur. Setiap peta fitur yang dikumpulkan sesuai dengan satu peta fitur dari lapisan sebelumnya. Dalam sebagian besar CNN, metode subsampling yang umum digunakan adalah max pooling. Dalam max pooling, output dari lapisan konvolusi dibagi menjadi beberapa grid kecil, dan nilai maksimum dari setiap grid diambil untuk membentuk matriks citra yang telah direduksi. Di sisi lain, average pooling membagi output menjadi beberapa grid kecil dan mengambil nilai rata-rata dari setiap grid pada citra untuk membentuk matriks yang telah direduksi[18]. Gambar 2.9. mengilustrasikan max pooling

### 2.9.3. Max Pooling

*Max pooling* adalah teknik dalam jaringan saraf konvolusional (*Convolutional Neural Network*, CNN) yang digunakan untuk mengurangi dimensi dan menghilangkan redundansi dalam fitur peta (*feature map*). Max pooling bekerja dengan cara memindai input menggunakan jendela berukuran tertentu dan memilih nilai maksimum dari setiap area yang dilintasi jendela tersebut. Proses ini membantu dalam menyaring fitur yang paling menonjol dan mengurangi jumlah parameter, sehingga meningkatkan efisiensi dan kinerja jaringan.[18]

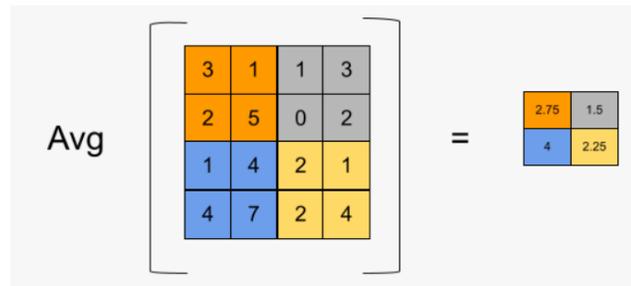
### 2.9.5. Average Pooling

*Average pooling* adalah teknik dalam jaringan saraf konvolusional yang digunakan untuk mengurangi dimensi peta fitur (*feature map*) dengan cara mengambil nilai rata-rata dari elemen-elemen dalam jendela *pooling*. Ini membantu dalam mereduksi data sambil mempertahankan informasi penting dari peta fitur, sehingga mempermudah proses pengolahan selanjutnya dan mengurangi kompleksitas model.[18]



**Gambar 2.9. Ilustrasi Max Pooling**

(Sumber <https://analyticsindiamag.com>)

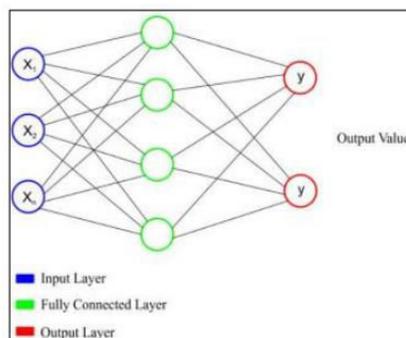


**Gambar 2.10. Ilustrasi Average Pooling**

(Sumber <https://analyticsindiamag.com>)

### 2.9.6. Fully-Connected Layer

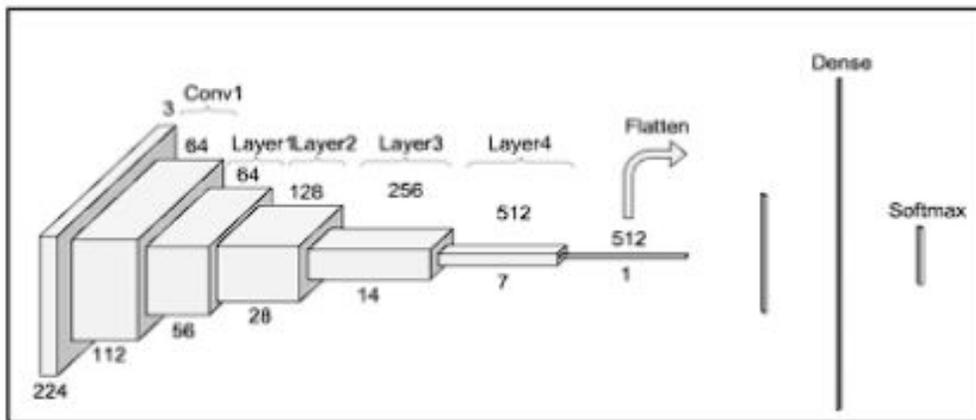
*Fully-Connected Layer* adalah lapisan di mana setiap neuron aktivasi dari lapisan sebelumnya terhubung dengan semua neuron di lapisan berikutnya, mirip dengan jaringan saraf biasa. Sebelum dihubungkan ke lapisan ini, aktivasi dari lapisan sebelumnya diubah menjadi data satu dimensi. Fully-Connected Layer biasanya digunakan dalam *Multi Layer Perceptron* (MLP) untuk melakukan transformasi dimensi data agar dapat diklasifikasikan secara linear. Perbedaan antara fully connected layer dan lapisan konvolusi terletak pada pola koneksi neuron. Pada lapisan konvolusi, neuron terhubung hanya ke area tertentu pada input (*field of view*), sedangkan pada *fully connected layer*, setiap *neuron* terhubung dengan semua neuron di lapisan sebelumnya. Meskipun demikian, kedua jenis lapisan ini menggunakan operasi perkalian matriks (*dot product*), sehingga esensinya memiliki fungsi yang serupa dalam proses transformasi dan pemrosesan data[18]. Ilustrasi *fully connected layer* yang ditunjukkan pada Gambar 2.6.



**Gambar 2.6. Ilustrasi Fully Connected Layer**

## 2.10. Residual Network

Residual Network (ResNet) memenangkan ILSVRC pada tahun 2015 dengan arsitektur yang memungkinkan pembuatan jaringan saraf dengan 50 lapisan, mengurangi kompleksitas dibandingkan dengan VGGNet. Tingkat kesalahan yang dicapai oleh ResNet ini adalah 3,57%, bertujuan untuk mengungguli tingkat kinerja manusia pada dataset yang diberikan. Kelebihan model ResNet dibandingkan dengan model struktural CNN lainnya adalah tampilan model ini tidak berkurang meskipun desainnya semakin maju. Diantaranya ada beberapa layer yang menghasilkan aspek gambar yang lebih sederhana[19]. Arsitektur ini menjadi terkenal karena memperkenalkan blok residu (residual block) yang memungkinkan pelatihan jaringan yang lebih dalam dengan mengatasi masalah degradasi performa yang sering terjadi pada jaringan yang lebih dalam. ResNet memiliki blok residual yang diperkenalkan oleh Kaiming He et al. dalam makalah "Deep Residual Learning for Image Recognition". Konsep ini dirancang untuk mengatasi beberapa masalah yang dihadapi oleh jaringan saraf yang sangat dalam, seperti masalah gradien yang menghilang (vanishing gradient) dan kesulitan dalam pelatihan.[20]



**Gambar 2.2. Ilustrasi Arsitektur ResNet50**

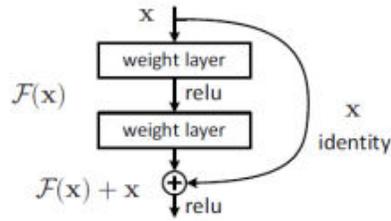
ResNet, atau Deep Residual Network, adalah struktur Convolutional Neural Network (CNN) yang diusulkan oleh He pada tahun 2016. Desain ini bertujuan untuk mengatasi masalah pelatihan jaringan yang sangat dalam dengan menerapkan jalur pintasan atau shortcut connections. ResNet memiliki keunggulan dalam menjaga kinerja model bahkan ketika arsitekturnya semakin kompleks, dan dapat menghasilkan representasi yang lebih kuat dari fitur gambar. Dengan pendekatan ini, ResNet berhasil mengatasi kendala pemahaman mendalam yang

terbatas pada beberapa lapisan tertentu dalam jaringan.[19] seperti digambarkan pada Gambar 2.2.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 8$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

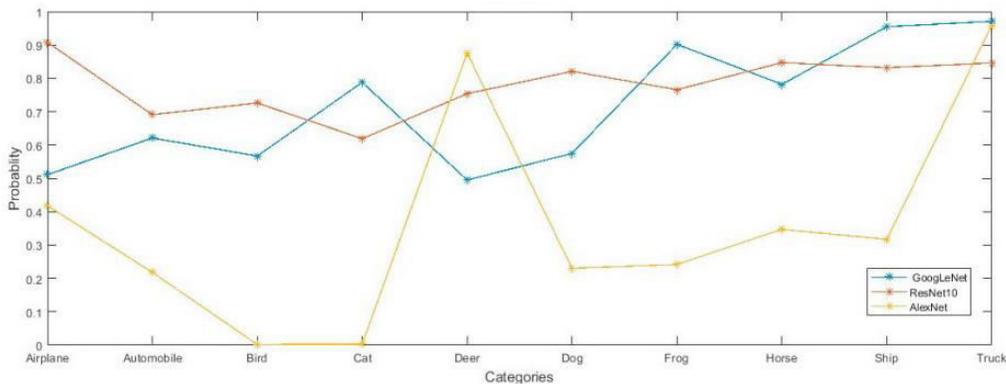
**Gambar 2.3. Arsitektur Layer ResNet50**

ResNet50 adalah sebuah Convolutional Neural Network (CNN) yang terdiri dari serangkaian komposisi lapisan. Setiap komposisi dimulai dengan lapisan conv1 yang menggunakan filter 7x7 dan menghasilkan output berukuran 112x112. Selanjutnya, lapisan conv2 melibatkan max pooling 3x3, diikuti oleh batch normalization, ReLU, dan convolution 3x3 yang diulang sebanyak 3 kali untuk menghasilkan output berukuran 56x56. Kemudian, lapisan conv3 terdiri dari batch normalization, ReLU, dan convolution 3x3 yang diulang 4 kali untuk menghasilkan output berukuran 28x28. Proses berlanjut dengan lapisan conv4 yang menggunakan batch normalization, ReLU, dan convolution 3x3 yang diulang 6 kali, menghasilkan output berukuran 14x14. Selanjutnya, lapisan conv5 terdiri dari batch normalization, ReLU, dan convolution 3x3 yang diulang 3 kali, menghasilkan output berukuran 7x7. Terakhir, dilakukan average max pooling untuk menghasilkan output berukuran 1x1 yang kemudian diumpankan ke lapisan softmax untuk menghasilkan prediksi akhir. ResNet50 menggunakan struktur ini untuk mengembangkan representasi-fitur yang semakin kompleks dari gambar input, dengan tujuan akhir untuk klasifikasi objek melalui softmax pada lapisan terakhir.[15]



**Gambar 2.4. Struktur Model ResNet**

Pada penelitian Neha Sharma[21] memungkinkan untuk mengevaluasi apakah probabilitas yang baik dan berguna diberikan kepada hasil yang benar, tetapi juga untuk mengamati secara kualitatif konsistensi hasil untuk setiap kategori, yaitu diharapkan bahwa untuk setiap kategori, 10 probabilitas teratas tidak bervariasi secara signifikan. Ini dicapai dengan meranking hasil yang diperoleh oleh pengklasifikasi. Semakin tinggi peringkat, semakin baik klasifikasinya. Idealnya, kelas yang benar akan berada di posisi pertama. Hitung rata-rata dan deviasi standar untuk setiap kategori. Rata-rata rendah sesuai dengan posisi yang lebih tinggi dalam peringkat, sementara standar deviasi rendah adalah bukti konsistensi produksi untuk instance yang berbeda dari kategori yang sama. Ini juga memungkinkan Anda untuk menangkap instance terbaik dan terburuk dari setiap kategori yang kami gunakan untuk menganalisis kemungkinan alasan dari hasil yang diamati. Akhirnya, kami dapat menyimpulkan dari hasil yang diperoleh bahwa kinerja rata-rata dari ketiga jaringan ini. Dari hasil yang diperoleh bahwa kinerja rata-rata dari ketiga jaringan ini pada dataset CIFAR10 adalah sebagai berikut: untuk AlexNet- 36,12%, untuk GoogLeNet- 71,67%, dan untuk ResNet50- 78,10% ditemukan.



**Gambar 2.5. Perbandingan Hasil Dari ke-Tiga Jaringan**

### 2.10.1. Zero Padding

Zero padding adalah operasi penambahan nilai 0 di kolom serta baris matriks sehingga matriks citra tadi menjadi lebih besar yang akan berakibat citra tersebut semakin halus [15]. Ilustrasi zero padding yang ditunjukkan pada Gambar 2.7.

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

Gambar 2.7. Ilustrasi Zero Padding

### 2.10.2. Batch Normalization

Batch normalization merupakan teknik untuk melatih deep neural network dengan melakukan standardizes input layer untuk setiap mini batch. Batch normalization digunakan buat mempercepat training jaringan menggunakan internal covariate shift. Internal covariate shift mengacu pada perubahan distribusi input pada learning system. batch normalization bermanfaat untuk mempercepat performa serta kestabilan pada jaringan[15]. Berikut persamaan batch normalization yang ditunjukkan pada persamaan (2.8), (2.9), (2.10), (2.11).

$$\mu_B^{(k)^2} = \frac{1}{m} \sum_{i=1}^m x_i^{(k)} \quad (2.8).$$

$$\sigma_B^{(k)^2} = \frac{1}{m} \sum_{i=1}^m (x_i^{(k)} - \mu_B^{(k)^2}) \quad (2.9).$$

$$x_i^{(k)} = \frac{x_i^{(k)} - \mu_B^{(k)^2}}{\sqrt{\sigma_B^{(k)^2} + \epsilon}} \quad (2.10).$$

$$y_i^{(k)} = \gamma^{(k)} x_i^{(k)} + \beta^{(k)} = BN_{\gamma^{(k)}, \beta^{(k)}}(x_i^{(k)})$$

(2.11).

Keterangan:

$\mu_B^{(k)}$  : mean mini batch

$\sigma_B^{(k)}$  : varians mini batch

$\varepsilon$  : numerical stability nilai epsilon (konstanta bernilai 0.00001)

$x_i^{(k)}$  : normalize

$y_i^{(k)}$  : output batch normalization

### 2.10.3. *ReLU*

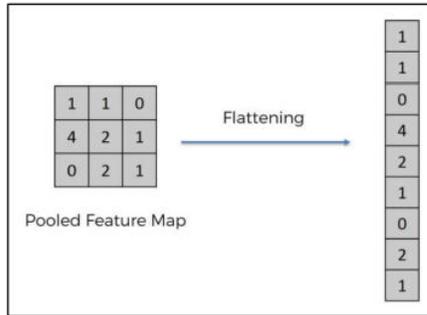
*Rectified Linear Unit* (ReLU) adalah sebuah fungsi aktivasi yang memiliki perhitungan sederhana. Dalam proses forward dan backward melalui ReLU, hanya digunakan kondisi if untuk pengolahan data. Jika elemen bernilai negatif, nilai tersebut akan diset menjadi 0, tanpa melibatkan operasi eksponensial, perkalian, atau pembagian[18].

Relu pada intinya hanya membuat pembatas pada bilangan nol, artinya apabila  $x \leq 0$  maka  $x = 0$  dan apabila  $x > 0$  maka  $x = x$  [11] Berikut persamaan fungsi aktivasi relu yang ditunjukkan pada persamaan (2.12).

$$f(x) = \begin{cases} x & (x \geq 0) \\ 0 & (x < 0) \end{cases} \quad (2.12).$$

### 2.10.4. *Flatten*

Flatten adalah operasi yang mengganti matriks menjadi vektor satu dimensi. Proses flattening mengubah feature map yang sudah diperoleh dari layer sebelumnya menjadi vektor satu dimensi agar feature map tadi dapat diklasifikasikan menggunakan fully connected layer serta softmax [11]. Proses flattening dapat dilihat pada Gambar 2.10.



**Gambar 2.12. Proses Flattening**

### 2.11. *Softmax*

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[15]. Berikut Rumus 2.13 merupakan persamaan softmax activation.

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad (2.13).$$

### 2.12. Confusion Matrix

Confusion Matrix merupakan metode yang digunakan untuk mengukur performa suatu model. Dalam proses pengukuran kinerja model klasifikasi[18]. Confusion matrix tersebut dapat diwakili dalam tabel yang terdapat pada Gambar 2.11.

		<i>Predicted class</i>		Total
		Yes	No	
<i>Actual class</i>	Yes	TP	FN	P
	No	FP	TN	N
	Total	P	N	P+N

**Gambar 2.13. Tabel Confusion Matriks**

terdapat beberapa istilah umum yang digunakan, yaitu:

True Positive (TP) - data positif yang diprediksi benar.

True Negative (TN) - data negatif yang diprediksi benar.

False Positive (FP) - data negatif namun diprediksi sebagai data positif.

False Negative (FN) - data positif namun diprediksi sebagai data negatif.

### 2.13. Optimizer Adam

Adam (*Adaptive Moment Estimation*) *Optimizer* adalah algoritma yang digunakan dalam pelatihan model jaringan saraf tiruan (neural network) untuk mengoptimalkan dan memperbaiki bobot yang ada dalam jaringan. Tujuannya adalah untuk meningkatkan efisiensi pembelajaran model serta mencapai kinerja yang lebih baik. Algoritma Adam menerapkan konsep momentum dan perubahan adaptif (adaptive learning rate) untuk melakukan pembaruan bobot dalam jaringan secara efektif.[18].

### 2.14. Akurasi , Precision, Recall, dan F-Measure Akurasi

Akurasi merupakan persentase data uji yang diklasifikasikan dengan benar. Hal ini digunakan sebagai parameter untuk mengukur keakuratan suatu model dalam klasifikasi[18]. Persamaan untuk menghitung akurasi dapat dinyatakan sebagai berikut:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN}$$

Presisi adalah ukuran yang menggambarkan seberapa tepat suatu model dalam memprediksi kejadian positif dalam serangkaian kegiatan prediksi. Untuk menghitung tingkat

presisi dalam prediksi suatu kejadian, dapat menggunakan persamaan yang menghitung jumlah prediksi positif yang benar dibagi dengan total jumlah prediksi positif yang dilakukan[18].

$$Precision = \frac{TP}{TP+FP}$$

Recall merupakan ukuran kelengkapan yang menggambarkan persentase tupel positif yang berhasil diklasifikasikan sebagai kelas positif dari seluruh tupel yang sebenarnya positif dalam dataset. Untuk menghitung recall[18]. Kita dapat menggunakan persamaan:

$$Recall = \frac{TP}{TP+FN}$$

Formula ini memberikan gambaran seberapa baik model dalam mengidentifikasi semua kasus positif yang sebenarnya ada. Recall juga dikenal sebagai sensitivity atau true positive rate (TPR).

F-measure merupakan metrik evaluasi dalam information retrieval (temu kembali) yang menggabungkan recall dan precision menjadi satu nilai tunggal yang mencerminkan keseimbangan antara kedua metrik tersebut[18]. Untuk menghitung F-measure, dapat menggunakan persamaan:

$$F - Measure = \frac{2*Precision*Recall}{Precision+Recall}$$