

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Tinjauan pustaka adalah komponen penting dalam sebuah penelitian yang mencakup referensi-referensi asli, terbaru, dan relevan. Di dalam tinjauan pustaka, disajikan berbagai bahan penelitian temuan, teori, yang akan memfasilitasi penulis dalam merumuskan kerangka konseptual atau pemikiran yang akan diadopsi dalam penelitian ini. Berikut adalah beberapa penelitian terdahulu.:

Penelitian sebelumnya yang dilakukan Nur Fadlia, dkk[11] dengan judul “Klasifikasi Jenis Kendaraan Menggunakan Metode *Convolutional neural network* (CNN)”. Penelitian ini dilakukan karena dalam pencegahan kemacetan dengan menetapkan jalur-jalur khusus yang hanya boleh dilalui kendaraan roda dua, roda empat, atau lebih masih banyak kendaraan yang berjalan tidak sesuai jalurnya. Citra yang digunakan untuk pelatihan model sebanyak 90 citra yang terdiri dari 30 citra sepeda, 30 citra mobil dan 30 citra motor. Terdapat 10 citra berbeda dari dataset pelatihan model untuk dataset pengujian. Model CNN yang digunakan berhasil melakukan klasifikasi dengan akurasi sebesar 73,33% terhadap dataset uji.

Penelitian sebelumnya yang dilakukan Ghazanfar Latif, dkk[12] dengan judul “*Deep learning Utilization in Agriculture: Detection of Rice Plant Diseases Using an Improved CNN Model*”. Penelitian tersebut dilakukan karena nasi merupakan makanan yang paling banyak dikonsumsi secara global. Belakangan ini terdapat penurunan produksi nasi karena penyakit pada padi yang akan berpengaruh pada kualitas bulir nasi. Citra yang digunakan sebanyak 1733 citra untuk 6 kelas sebagai dataset pelatihan model dan memiliki 434 citra untuk 6 kelas sebagai dataset pengujian model. Model CNN yang digunakan adalah *transfer*

*learning* dari VGG-19 dengan pendekatan yang dimodifikasi. Akurasi rata-rata tertinggi yang dihasilkan adalah 96.08% terhadap dataset uji.

Penelitian sebelumnya yang dilakukan Toton Dwi Antoko, dkk[13] dengan judul “Klasifikasi Buah Zaitun Menggunakan *Convolution Neural Network*”. Penelitian ini dilaksanakan karena minyak yang dihasilkan dari buah zaitun merupakan produk utama yang berasal dari buah zaitun, dengan konsumsinya yang telah meningkat dua kali lipat dalam dua dekade terakhir. Akibatnya, sektor zaitun menghadapi tantangan besar untuk memenuhi tingginya permintaan pasar. Dalam penelitian ini, digunakan 136 citra dengan enam kategori. Citra-citra tersebut melalui tahap *preprocessing* berupa pemotongan, di mana setiap citra dipotong untuk mendapatkan buah zaitun individu yang berbeda, menghasilkan total 267 citra. Akurasi yang diperoleh model CNN dalam penelitian ini mencapai 92%.

Penelitian sebelumnya yang dilakukan Stifani Napitu, dkk[14] dengan judul “Klasifikasi Buah Jeruk Segar dan Busuk Berdasarkan RGB dan HSV Menggunakan Metode KNN”. Penelitian tersebut dilakukan karena tingginya tingkat permintaan yang meminta para petani untuk dapat mengklasifikasikan tingkat kesegaran. Citra yang digunakan adalah jeruk dengan total 146 citra untuk data *training* dan total 88 citra untuk data *testing*. Hasil akurasi akhir yang didapat adalah 88,95%.

Penelitian sebelumnya yang dilakukan Ratna Dhamayanti, dkk[15] dengan judul “Penggunaan *Deep learning* Dengan Metode *Convolutional neural network* Untuk Klasifikasi Kualitas Sayur Kol Berdasarkan Citra Fisik”. Penelitian ini dilakukan sebagai respons terhadap penemuan kecacatan dalam bentuk fisik pada beberapa sayur kol. Akibatnya, dibutuhkan proses seleksi ulang yang bisa memakan waktu cukup lama jika dilakukan secara manual oleh manusia. Dataset terdiri dari 210 citra yang digunakan untuk pelatihan model dan 60 citra untuk pengujian model. Hasil akurasi akhir yang didapat dari model yang sudah dilatih di penelitian tersebut adalah 80% terhadap data pengujian.

Tabel 2. 1 Ringkasan Penelitian Sebelumnya

No	Judul	Tahun	<i>Compare</i>	<i>Contrast</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
1	Klasifikasi Jenis Kendaraan Menggunakan Metode <i>Convolutional neural network</i> (CNN).	2019	Penelitian ini melakukan klasifikasi dengan menggunakan metode CNN.	Penelitian ini menggunakan citra kendaraan bermotor sebagai dataset	Pada penelitian ini perlu menurunkan <i>loss</i> yang dihasilkan.	Penelitian ini menggunakan metode CNN menggunakan sebanyak 120 data citra sebagai dataset. Dataset dibagi menjadi 3 kelas yaitu 30 citra mobil, 30 citra motor dan 30 citra sepeda. Masing masing kelas memiliki masing masing tambahan 10 citra untuk pengujian. Proses pelatihan model menggunakan lapisan konvolusi, lapisan <i>pooling</i> , lapisan <i>dropout</i> , lapisan <i>flatten</i> dan lapisan <i>dense</i> .	Penelitian ini berhasil menggunakan CNN untuk mengklasifikasikan jenis kendaraan dengan tingkat akurasi sebesar 73,33%. Selama proses pelatihan, model CNN mencapai tingkat akurasi sebesar 94,44% dengan <i>loss</i> sebesar 0,171811.

No	Judul	Tahun	<i>Compare</i>	<i>Contrast</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
							Namun, dalam tahap pengujian, terjadi peningkatan nilai <i>loss</i> yang relatif tinggi, yaitu sebesar 0,9727243, dengan akurasi sebesar 73,33%.
2	<i>Deep learning Utilization in Agriculture: Detection of Rice Plant Diseases Using an Improved CNN Model.</i>	2022	Penelitian ini pengklasifikasian menggunakan metode CNN dengan basis transfer learning dari VGG-19 lalu dimodifikasi.	Menggunakan citra kelainan pada daun padi sebagai dataset.	Penelitian ini perlu pengoptimalan teknik <i>deep learning</i> yang dilakukan.	Pada penelitian ini menggunakan metode CNN dengan arsitektur VGG19 yang dimodifikasi lalu dibandingkan dengan model CNN yang lain selama proses pemodifikasian untuk mendapatkan hasil yang	Penelitian ini mendapatkan akurasi tertinggi rata-rata sebesar 96.08% dari model yang dimodifikasi.

No	Judul	Tahun	<i>Compare</i>	<i>Contrast</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
						optimal.	
3	Klasifikasi Buah Zaitun Menggunakan <i>Convolution Neural Network</i> .	2021	Penelitian ini melakukan klasifikasi dengan menggunakan metode CNN.	Penelitian ini menggunakan citra buah zaitun sebagai dataset.	Penelitian ini perlu memperbanyak dataset yang digunakan.	Pada penelitian ini menggunakan CNN untuk mengklasifikasikan buah zaitun dengan komputasi yang lebih cepat dengan membandingkan beberapa arsitektur model. Setelah pengumpulan data dilakukan Augmentasi data dengan fungsi rotasi untuk memperbanyak data dengan variasi gambar yang dirotasi dari 0 sampai 360 derajat.	Penelitian ini mendapatkan akurasi sebesar 92%. Perbandingan model dengan arsitektur berbeda mendapatkan hasil rata-rata yang lebih tinggi. Namun, kekurangan pada model yang dibuat peneliti adalah tidak adanya dataset untuk buah zaitun jenis <i>ocal</i> .
4	Klasifikasi Buah Jeruk Segar dan Busuk	2023	Penelitian ini melakukan klasifikasi menggunakan	Penelitian menggunakan citra Jeruk sebagai	Penelitian ini perlu menambahkan variasi citra	Pada penelitian ini, KNN dapat melakukan klasifikasi antara buah jeruk yang matang dan	Penelitian ini mendapatkan akurasi sebesar 88,95% yang di

No	Judul	Tahun	<i>Compare</i>	<i>Contrast</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
	Berdasarkan RGB dan HSV Menggunakan Metode KNN		metode KNN	dataset	pada dataset	busuk dengan memanfaatkan citra warna dari warna kulit jeruk melalui dua pemrosesan citra yaitu segmentasi lalu ekstraksi ciri.	dapat dari rata-rata seluruh nilai k.
5	Penggunaan <i>Deep learning</i> Dengan Metode <i>Convolutional neural network</i> Untuk Klasifikasi Kualitas Sayur Kol Berdasarkan Citra Fisik	2021	Penelitian ini melakukan klasifikasi dengan menggunakan metode CNN	Penelitian ini menggunakan citra sayur kol sebagai dataset	Pada penelitian ini perlu menambahkan <i>preprocessing</i> data.	Pada penelitian ini, peneliti menggunakan metode CNN dengan tiga algoritma performansi berbeda yaitu <i>Stochastic Gradient Descent (SGD)</i> , <i>Adaptive Moment (Adam)</i> dan <i>Root Mean Square Propagation(RMSProp)</i> .	Penelitian ini mendapatkan akurasi tertinggi dengan performansi <i>ADAM</i> yaitu 80% untuk data <i>Test</i> dan 73% untuk data <i>Train</i> .

## 2.2 Landasan Teori

### 2.2.1 Kualitas Buah Pisang

Kualitas buah pisang dipengaruhi oleh beberapa faktor seperti pH, keasaman yang dapat dititrasi, padatan terlarut, indeks kematangan, kekenyalan, warna kulit, dan kadar gula, yang membantu dalam menentukan baik dan buruknya kualitas pisang[16]. Warna pisang dapat digunakan untuk memprediksi atribut tekstur seperti kekenyalan dan tingkat kekerasannya[17]. Contohnya, perpindahan warna dari hijau ke kuning pada kulit buah menandakan pemecahan klorofil dan konversi gula menjadi pati di dalam daging buah, sehingga menghasilkan buah yang lebih lembut dan manis[18].

### 2.2.2 Kualitas Buah Jeruk

Jeruk dikenal dengan kandungan vitamin C dan manfaatnya bagi kesehatan. Kualitas jeruk ditentukan oleh parameter seperti padatan terlarut, keasaman yang dapat dititrasi, dan hubungan di antara keduanya, yang membantu memperkirakan tingkat kematangan fisiologis buah[19]. Warna juga merupakan indikator kualitas yang penting untuk jeruk, yang mencerminkan kandungan karotenoid yang ditandai dengan warna kuning cerah hingga merah dan pigmen lain yang memiliki manfaat bagi kesehatan[20][21].

### 2.2.3 *Artificial Intelligence*

*Artificial Intelligence* atau kecerdasan buatan adalah studi tentang bagaimana membuat komputer bekerja tugas-tugas cerdas yang di masa lalu, hanya dapat dilakukan oleh manusia[22]. Kecerdasan Buatan (*Artificial Intelligence*) mampu melaksanakan tugas-tugas yang biasanya membutuhkan kecerdasan manusia kini dapat dilakukan oleh teknologi ini. Sistem ini mampu mengambil keputusan dengan menganalisis dan memanfaatkan data yang tersedia dalam sistem. Proses dalam AI mencakup pembelajaran,

penalaran setelah itu koreksi diri. Mekanisme ini serupa dengan analisis yang dilakukan manusia sebelum membuat keputusan. Cara kerja ini mirip dengan analisis yang dilakukan manusia sebelum mengambil keputusan.[23].

#### 2.2.4 *Machine Learning*

*Machine learning* adalah cabang ilmu komputer yang luas dengan tujuan memungkinkan komputer untuk "belajar" tanpa perlu diprogram secara langsung. [24]. *Machine learning* juga dapat didefinisikan sebagai algoritma yang bertujuan untuk menemukan dan menerapkan pola-pola dalam data. Algoritma ini menggunakan teknik-teknik statistik untuk mengidentifikasi pola-pola tersebut. Seringkali, data yang dianalisis berukuran besar dan tidak hanya berupa teks, tetapi juga dapat mencakup gambar, audio, video, atau aktivitas pengguna saat *browsing* di internet atau mengakses informasi secara online.[25].

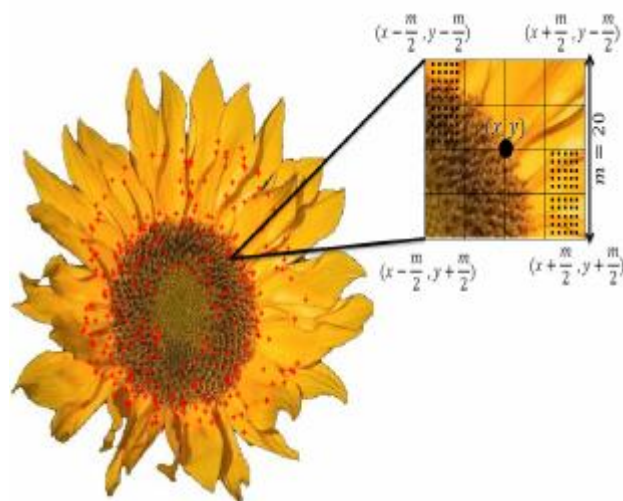
#### 2.2.5 *Deep learning*

*Deep learning* adalah salah satu cabang dari *machine learning* yang melakukan pembelajaran lebih mendalam melalui banyak lapisan atau *layer* [26]. Lapisan-lapisan tersebut terdiri dari kumpulan *node*. Sebuah *node* berfungsi sebagai tempat di mana perhitungan terjadi. *Node* input digabungkan dengan bobot (*weight*), kemudian jumlah dari input dan bobot tersebut melewati tahap yang disebut fungsi aktivasi *node*, yang menentukan sejauh mana sinyal diteruskan melalui jaringan, mempengaruhi hasil akhir. Perbedaan antara *Deep learning* dan *neural network* adalah jumlah *hidden layer* yang lebih banyak pada *Deep learning*. Jika terdapat lebih dari tiga lapisan (termasuk lapisan *input* dan *output*), maka itu memenuhi syarat sebagai "*deep*" *learning*. [27].



### 2.2.6 Features Extraction

Ekstraksi fitur merujuk pada metode pengolahan citra atau gambar yang bertujuan untuk menggambarkan karakteristik khusus dari suatu gambar sebelum dilakukan proses klasifikasi. Metode ini bertujuan untuk mempermudah pemisahan antara gambar-gambar berdasarkan ciri-ciri yang telah diambil. Ciri atau fitur ini direpresentasikan dalam bentuk urutan bilangan yang dapat digunakan untuk mengidentifikasi objek dalam gambar. Salah satu jenis fitur yang dapat dimanfaatkan adalah fitur warna. [28]. Contoh ilustrasi pengambilan ciri untuk dipelajari mesin dapat dilihat pada Gambar 2.1.

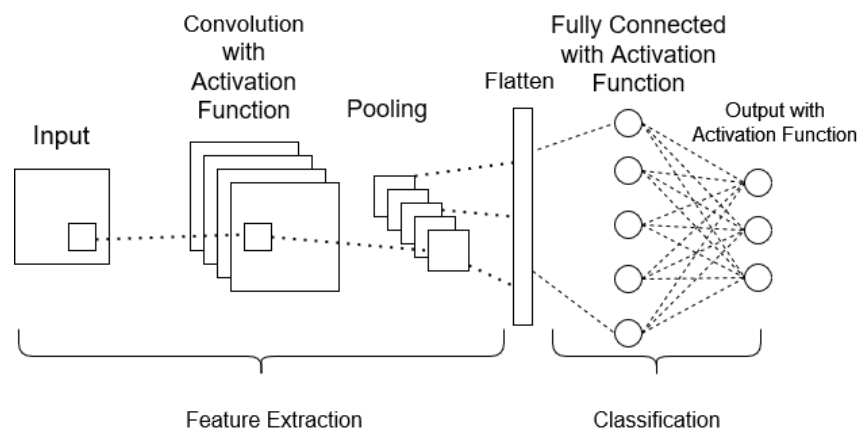


Gambar 2. 1 Pengambilan ciri pada citra [29].

### 2.2.7 Convolutional neural network

*Convolutional neural network* (CNN) adalah metode atau algoritma jaringan saraf atau *neural network* mendalam yang paling umum digunakan untuk menganalisis visual. *Convolutional neural network* menggunakan multilayer perceptron di mana setiap *neuron* atau *node* terhubung dengan semua neuron atau *node* di lapisan berikutnya. Namun, *Convolutional neural network* memiliki kemampuan untuk menemukan pola hierarki dalam data, menggabungkan piksel-

piksel yang lebih kompleks dari piksel-piksel yang lebih kecil dan sederhana. Oleh karena itu, *Convolutional neural network* memiliki performa yang sangat baik dalam hal keterhubungan dan kompleksitas piksel gambar. [30]. Berikut contoh arsitektur *Convolutional neural network* (CNN) pada Gambar 2.1



Gambar 2. 2 Arsitektur dan komponen CNN.

#### 2.2.4.1 Input Layer

*Input layer* berfungsi untuk menerima citra *input* yang akan diolah oleh model berbentuk matriks *pixel*. Jika diasumsikan bahwa citra masukan memiliki ukuran 64 x 64 piksel dan menggunakan model warna RGB (Merah, Hijau, Biru), maka citra *input* ini dapat dianggap sebagai sebuah *array* multidimensi dengan dimensi 64 x 64 x 3. Dimensi 3 ini mewakili tiga saluran warna, yaitu merah, hijau, dan biru. Dengan kata lain, citra input adalah kumpulan nilai piksel dalam tiga saluran warna yang membentuk citra berukuran 64 x 64 piksel[31].

#### 2.2.4.2 Convolutional Layer

*Convolutional Layer* bertugas untuk mengekstrak fitur-fitur penting dari citra input dengan menggunakan sejumlah

filter atau kernel. Setiap kernel bergerak secara berulang-ulang melalui citra, melakukan operasi konvolusi untuk mendeteksi pola dan fitur seperti tepi, tekstur, atau bentuk[31][32].

#### 2.2.4.3 Pooling Layer

Tujuan dari *Pooling Layer* adalah untuk menghilangkan informasi yang berlebihan dan mempertahankan yang lebih bermanfaat. Menentukan fitur mana yang informatif dan mana yang tidak relevan/berlebihan sangat penting dalam metode *Pooling layer*. *Max pooling* dan *Average Pooling* dikenal sebagai metode yang efektif dan banyak digunakan dalam praktik karena kesederhanaan dari metode *Max Pooling* dan *Average Pooling* [33].

#### 2.2.4.4 Flatten Layer

*Flatten Layer* berfungsi mengubah input menjadi *vector* (array satu dimensi) satu dimensi. Citra yang telah diolah dalam semua *layer* sebelumnya diubah menjadi vektor satu dimensi untuk diproses oleh lapisan-lapisan berikutnya[34].

#### 2.2.4.5 Activation Function

*Activation function* atau fungsi aktivasi memiliki peran penting dalam kemampuan jaringan saraf atau *neuron* yang sangat dalam untuk membedakan data. Fungsi ini juga merupakan salah satu faktor utama yang mendorong kemajuan dalam pengembangan jaringan saraf. Fungsi Aktivasi digunakan untuk memperkenalkan ketidaklinearan ke dalam model, yang memungkinkan model menganalisis hubungan yang lebih kompleks dalam data[35]. Fungsi aktivasi yang umum digunakan termasuk fungsi tangen hiperbolik (*tanh*), fungsi

sigmoid, dan unit linier yang diperbaiki yaitu Rectified Linear Unit (ReLU) [36]. Fungsi-fungsi ini, meskipun efektif, dapat mengalami masalah seperti gradien yang menghilang, yang dibantu oleh ReLU untuk dikurangi.

Penggabungan ReLU dalam arsitektur CNN telah terbukti berkontribusi pada proses pelatihan yang lebih cepat dikarenakan kesederhanaan operasi matematis dan lebih efektif, menjadikannya pilihan yang lebih disukai untuk berbagai tugas seperti segmentasi semantik, pengenalan gerakan tangan, dan klasifikasi gambar[37]. Berikut rumus ReLU yang mengubah masukan negatif menjadi 0 (nol) yang membuat ReLU lebih efektif dapat dilihat pada persamaan 2,1 [38].

$$ReLU(x) = \max(0, x) \quad (2,1)$$

#### 2.2.4.6 Fully Connected Layer

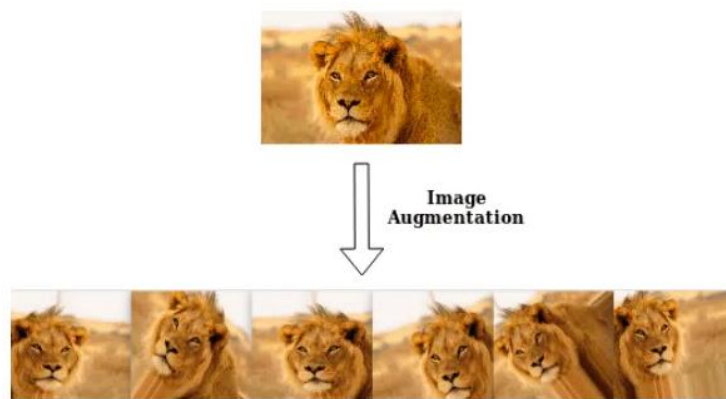
Pada *Fully Connected Layer*, semua *node (neuron)* dari lapisan sebelumnya terhubung satu sama lain dengan node di lapisan berikutnya. Pada setiap lapisan tersembunyi, terdapat fungsi aktivasi, yang umumnya adalah ReLU, dan hal yang sama berlaku pada lapisan keluaran, di mana, dalam hal mengklasifikasikan sesuatu (melibatkan lebih dari dua label atau kategori), *Softmax* biasanya digunakan. Tujuan utama dari lapisan terhubung penuh adalah memproses data untuk memungkinkan klasifikasi. Hasil keluaran dari lapisan terhubung penuh adalah probabilitas terkait dengan kategori (ketika menggunakan *Softmax*)[32].

#### 2.2.4.7 Output Layer

*Output Layer* menghasilkan output prediksi yang sesuai dengan tugas yang diberikan, seperti klasifikasi objek dalam kasus pengenalan gambar[32].

#### 2.2.8 Data Augmentation

*Data Augmentation* adalah strategi yang digunakan untuk mencegah *overfitting* dalam *machine learning*. Teknik-teknik lain seperti *dropout* dan *weight penalties* pada bobot juga digunakan untuk tujuan serupa, tetapi *Data Augmentation* memberikan cara yang lebih intuitif untuk mengerti bagaimana menghasilkan data tambahan yang tetap mempertahankan label data asli. *Data Augmentation* dapat digambarkan sebagai kumpulan algoritma yang digunakan untuk membuat data tambahan dari data yang sudah ada. Data tambahan ini biasanya berisi perubahan kecil pada data yang seharusnya tidak memengaruhi prediksi model[39]. Contoh dari *Data Augmentation* yang ada pada citra dapat terlihat pada Gambar 2.2



Gambar 2. 3 Contoh citra singa yang di augmentasi [40].

Gambar yang diunggah menunjukkan proses augmentasi gambar (*image augmentation*) pada gambar seekor singa. Augmentasi gambar adalah teknik yang digunakan dalam pembelajaran mesin dan pengolahan citra untuk meningkatkan

variasi data pelatihan dengan membuat berbagai modifikasi pada gambar asli. Teknik ini membantu model menjadi lebih generalis dan meningkatkan kinerjanya pada data yang belum pernah dilihat sebelumnya. Salah satu alat yang sering digunakan untuk augmentasi gambar adalah ImageDataGenerator dari Keras. Berikut adalah beberapa fungsi yang ada dalam ImageDataGenerator:

1. *Rotation Range*: Menentukan rentang rotasi gambar, memungkinkan model untuk lebih baik mengenali objek dari berbagai sudut.
2. *Shift Range*: Mengontrol rentang pergeseran horizontal dan vertikal, membantu model mengenali objek yang mungkin muncul di berbagai posisi.
3. *Shear Range*: Mengatur rentang pemotongan gambar, berguna untuk memperkenalkan variasi dalam bentuk objek.
4. *Zoom Range*: Menentukan rentang *zoom*, membantu model mengenali objek dengan skala yang berbeda.
5. *Horizontal Flip*: Memungkinkan atau tidak *flip* horizontal, menciptakan variasi tambahan dalam orientasi objek.
6. *Fill Mode*: Cara mengisi *pixel* yang mungkin muncul setelah augmentasi.

Dengan menggunakan fungsi-fungsi ini, ImageDataGenerator dapat menghasilkan berbagai versi modifikasi dari gambar asli, seperti yang terlihat pada gambar singa, untuk meningkatkan kemampuan generalisasi model pembelajaran mesin[39].

### 2.2.9 TensorFlow

*TensorFlow* adalah kerangka kerja komputasi yang dirancang untuk membangun model pembelajaran mesin atau *machine learning*. *TensorFlow* menawarkan berbagai *toolkit* yang

memungkinkan pengguna untuk membuat model pada tingkat abstraksi yang berbeda, serta dapat menjalankan grafik pada berbagai platform *hardware*, seperti CPU, GPU, dan TPU.[41].

*TensorFlow* dan alat terkait memungkinkan pengguna untuk dengan efisien memprogram dan melatih jaringan saraf dan model pembelajaran mesin lainnya, serta mendeploynya ke produksi. Algoritma inti TensorFlow ditulis dalam C++ yang sangat dioptimalkan dan CUDA (*Compute Unified Device Architecture*), platform komputasi paralel dan API yang diciptakan oleh NVIDIA. *TensorFlow* memiliki API yang tersedia dalam beberapa bahasa. API Python adalah yang paling lengkap dan stabil. Bahasa lain yang secara resmi didukung meliputi JavaScript, C++, Java, Go, dan Swift. Paket pihak ketiga juga tersedia untuk lebih banyak bahasa seperti C# dan Ruby.[42].

#### 2.2.10 Matplotlib

Matplotlib adalah salah satu pustaka Python yang paling populer untuk visualisasi data. Pustaka ini memungkinkan pembuatan plot 2 dimensi dari data dalam bentuk larik atau *array*. Selain itu, Matplotlib menyediakan API (Application Programming Interface) yang dapat digunakan dalam berbagai aplikasi GUI Python seperti PyQt, WxPython, atau Tkinter. Pustaka ini juga kompatibel dengan shell Python dan IPython, notebook Jupyter, dan server web.[43].

#### 2.2.11 Numpy

*Numpy* adalah proyek *open source* sumber terbuka yang bertujuan untuk memungkinkan komputasi numerik menggunakan *Python*. Dibuat pada tahun 2005, *Numpy* dibangun berdasarkan pustaka sebelumnya, yaitu *Numeric* dan *Numarray*. [44] *Numpy* dapat membantu melakukan operasi matematika

terutama ketika melakukan kalkulasi pada *vector*, *matrix* dan *tensor*[45]. Contoh menggunakan *library Numpy* dengan menggunakan *np.array* dan tanpa menggunakan *library numpy* untuk menghitung jumlah dari kedua *list* dapat dilihat pada gambar 2.3.

```
In [3]: a = [1, 2, 3]
       [q*2 for q in a]
Out[3]: [2, 4, 6]
```

```
In [4]: a = np.array([1, 2, 3])
       a * 2
Out[4]: array([2, 4, 6])
```

```
In [1]: a = [1, 2, 3]
       b = [4, 5, 6]
       [q+r for q, r in zip(a, b)]
Out[1]: [5, 7, 9]
```

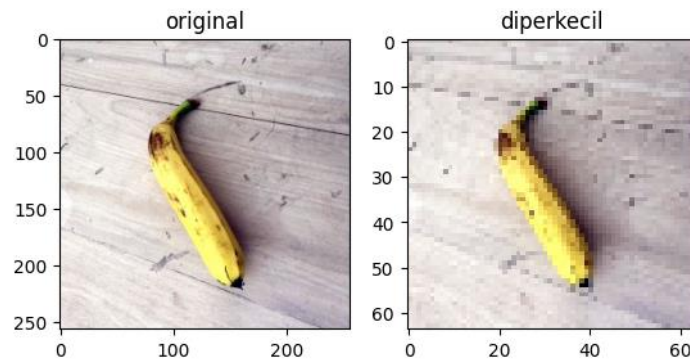
```
In [2]: a = np.array([1, 2, 3])
       b = np.array([4, 5, 6])
       a + b
Out[2]: array([5, 7, 9])
```

Gambar 2. 4 Penggunaan numpy untuk membantu operasi matematika [46].

### 2.2.12 *OpenCV*

*OpenCV* adalah *library open source* yang utamanya digunakan untuk aplikasi *Computer Vision*. *OpenCv* mempunyai fungsi yang banyak dan algoritma untuk deteksi objek, pengenalan wajah, segmentasi, serta berbagai aplikasi lainnya. Gambar dan *video real-time* dapat dimanipulasi untuk memenuhi berbagai kebutuhan menggunakan perpustakaan ini[47]. Contoh implementasi *library OpenCV* untuk memperkecil gambar berdasarkan satuan *pixel*

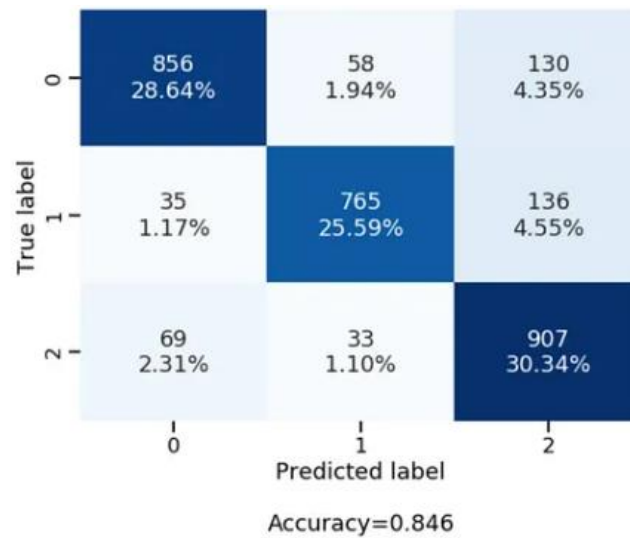




Gambar 2. 5 Memperkecil gambar berdasarkan satuan *pixel*.

### 2.2.13 Confusion matrix

*Confusion matrix* adalah salah satu metrik yang digunakan dalam proses klasifikasi di arsitektur CNN[48]. Contoh *confusion matrix* hasil dari pengklasifikasian model terhadap 3 kelas dengan keterangan berupa banyak data dalam beberapa kategori dapat dilihat pada gambar 2.5



Gambar 2. 6 *Confusion matrix heatmap* [49].

Penggunaan *Confusion matrix* memungkinkan pengukuran ketepatan dengan memeriksa nilai-nilai pada setiap kelas prediksi dan kelas aktual. [50]. Berikut visualisasi *Confusion matrix* pada Tabel 2.2

Tabel 2. 2 *Confusion matrix*

		Kelas Aktual	
		Negatif	Positif
Prediksi	Negatif	TN	FN
	Positif	FP	TP

*True Positives (TP)* adalah jumlah contoh yang sebenarnya positif dan telah diprediksi dengan benar sebagai positif oleh model. *True Negatives (TN)* adalah jumlah contoh yang sebenarnya negatif dan telah diprediksi dengan benar sebagai negatif oleh model. *False Positives (FP)* adalah jumlah contoh yang sebenarnya negatif tetapi telah salah diprediksi sebagai positif oleh model. *False Negatives (FN)* adalah jumlah contoh yang sebenarnya positif tetapi telah salah diprediksi sebagai negatif oleh model. Precision dan recall membantu dalam memahami keseimbangan antara kesalahan positif palsu (false positives) dan negatif palsu (false negatives). Misalnya, dalam deteksi kanker, recall yang tinggi sangat penting agar tidak ada kasus kanker yang terlewatkan (negatif palsu). Di sisi lain, dalam deteksi email spam, precision yang tinggi sangat penting agar email penting tidak ditandai sebagai spam (positif palsu) [48]. Pengukuran yang umum dilakukan untuk evaluasi yaitu:

1. *Precision* (Presisi): *Precision* mengukur seberapa akurat model dalam mengidentifikasi contoh yang benar-benar positif [50][51][52]. Kalkulasi dari nilai *Precision* diberikan di dalam persamaan 2,2

$$Precision = \frac{TP}{TP + FP} \quad (2,2)$$

2. *Recall*: *Recall* mengukur seberapa baik model dalam mengidentifikasi semua contoh positif yang ada [50][51][52]. Kalkulasi dari nilai *Recall* diberikan di dalam persamaan 2,3

$$Recall = \frac{TP}{TP + FN} \quad (2,3)$$

3. *F1-Score*: *F1-Score* adalah perbandingan rata-rata presisi dan *recall* yang diberi bobot. Skor F1 atau *F1 score* dianggap paling baik jika terdapat keseimbangan antara presisi dan *recall* dalam suatu sistem[52][53]. Kalkulasi dari nilai *F1 score* diberikan di dalam persamaan 2,4

$$F1\ score = \frac{2TP}{(2TP + FP + FN)} \quad (2,4)$$

#### 2.2.14 Accuracy

*Accuracy* atau akurasi dalam pembelajaran mesin atau *machine learning* mengacu pada kemampuan model untuk membuat prediksi atau klasifikasi yang benar. Ini adalah metrik fundamental dalam mengevaluasi kinerja algoritma pembelajaran mesin[54].

#### 2.2.15 Loss

*Loss* dalam pembelajaran mesin mengacu pada ukuran seberapa baik kinerja model pembelajaran mesin pada tugas yang diberikan. Hal ini biasanya diukur dengan fungsi kerugian, yang menghitung kesalahan antara *output* yang diprediksi dari model dan kebenaran yang sebenarnya. Fungsi kerugian yang berbeda digunakan tergantung pada masalah spesifik yang ditangani[55].