

BAB 2 DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian [8] membahas mengenai sistem prediksi penyakit daun teh berbasis *convolutional neural network* pada *smart phone* menggunakan *paas cloud*. Penelitian tersebut penerapan sistem prediksi penyakit secara *real-time* menggunakan CNN di *cloud Platform-as-a-Service* (PaaS). Dataset yang digunakan dalam penelitian ini tersedia di situs web data *Mendeley* dan terdiri dari tiga jenis penyakit daun teh, yaitu *Tea leaf blight*, *Tea red leaf spot*, dan *Tea red scab*. Dataset citra daun teh dibagi menjadi 90 data untuk pelatihan dan 30 data untuk validasi. Model CNN yang digunakan diimplementasikan menggunakan metode optimasi *Adam* dan *RMSprop*. Validasi silang dilakukan dengan menggunakan dataset pelatihan dan validasi, dan hasilnya menunjukkan akurasi sebesar 97%.

Penelitian [9] mengenai klasifikasi jenis penyakit teh menggunakan CNN. Dalam penelitian ini berfokus pada penggabungan arsitektur *GoogLeNet*, *Xception* dan *Inception-ResNet-v2*. Total dataset terkumpul sebanyak 4727 citra daun teh yang terdiri dari empat kelas yaitu penyakit melepuh, belalang daun, ulat looper dan daun sehat. Hasil klasifikasi terbaik untuk dua data kelas penyakit mencapai akurasi 98,09% dengan menggunakan *learning rate* 10^{-4} . Sementara itu, hasil menggunakan tiga kelas data memiliki akurasi 90,05% dengan menggunakan *learning rate* 10^{-4} . Untuk klasifikasi keempat kelas data, akurasi terbaik yang dicapai adalah 89,64% dengan menggunakan *learning rate* 10^{-4} . Penggunaan *learning rate* berperan dalam mengatur kecepatan pembelajaran model dalam menyelesaikan masalah.

Penelitian [10] mendeteksi dan mengidentifikasi penyakit daun teh tersebut menggunakan CNN. Dataset yang digunakan dalam penelitian klasifikasi penyakit daun teh ini diambil sendiri menggunakan *Realme AI powered 64Mp Quad Camera* di lingkungan cahaya alami dari kebun teh dengan total 2341 dataset. Pada lapisan konvolusi model ini, terdapat 32 filter dengan *kernel* berukuran 3×3

dan menggunakan fungsi aktivasi *Rectified Linear Units* (ReLU). Untuk mengatasi masalah dalam jaringan saraf, diterapkan *normalisasi batch*, *maximum pooling*, dan teknik *dropout* (regularisasi) sebesar 25% (0,25). Dengan menggunakan *dropout* sebesar 0,25, akurasi meningkat menjadi 90%. Hasil akhir dihitung menggunakan data validasi dan menunjukkan tingkat akurasi sekitar 95,938%.

Penelitian [11] membahas tentang klasifikasi penyakit daun teh dengan melakukan komparasi Algoritma *MobileNet* dan *Nasnet Mobile*. Menggunakan dataset dari publik yang diambil dari *kaggle* (Saikat Datta, 2022). Dataset terdiri dari 6 kelas yaitu, *algal spot*, *brown blight gray blight*, *helopeltis*, *red spot* dan *healthy*. Metode yang digunakan dalam penelitian ini yaitu mengkomparasi metode *MobilNet* dan *Nasnet Mobile* pada klasifikasi penyakit daun teh tersebut. Dari hasil komparasi menunjukkan bahwa *NasNet Mobile* mencapai akurasi sebesar 88%, sedangkan *MobileNet* mencapai akurasi sebesar 95%. Dengan demikian, berdasarkan hasil komparasi tersebut, dapat disimpulkan bahwa *MobileNet* memberikan tingkat akurasi yang lebih tinggi dalam klasifikasi penyakit pada daun teh.

Penelitian [12] ini mendeteksi dan menganalisis penyakit daun teh menggunakan *Deep Learning*. Dalam penelitian ini, digunakan metode pengolahan citra menggunakan CNN model *LeNet-5*. Terdapat empat kelas penyakit dalam penelitian ini yaitu *blister blight*, *red scab*, *red leaf spot* dan *leaf blight*. Hasil penelitian menunjukkan bahwa penyakit keropeng teh merah dapat diklasifikasikan dengan akurasi tertinggi, yaitu 94%. Sedangkan untuk penyakit lainnya, akurasi klasifikasi mencapai 84,93%. Dalam keseluruhan, hasil akurasi yang diperoleh adalah sebesar 90,23%.

Penelitian [13] pembahasan ini mengenai deteksi dan identifikasi daun teh menggunakan *Ax-RetinaNet*. Dataset yang digunakan dalam penelitian ini terdiri dari 520 citra daun teh yang dikelompokkan ke dalam empat kelas yaitu *tea algae leaf spot*, *tea bud blight*, *tea white scab* dan *tea leaf blight*. Metode *Ax-RetinaNet* menggunakan modul *attention* dan fusi fitur *X-module* untuk mendapatkan hasil deteksi dan identifikasi yang optimal. Hasil dari deteksi dan identifikasi menggunakan *Ax-RetinaNet* menunjukkan nilai mAP (*mean Average Precision*)

sebesar 93,83% dan skor F1 sebesar 0,954. Hal ini menunjukkan keefektifan dan keakuratan metode Ax-RetinaNet dalam deteksi penyakit pada daun teh.

Tabel 2. 1 Penelitian sebelumnya.

No	Judul	Nama Peneliti	Data	Metode	Perbedaan
1	<i>Convolutional neural network-based tea leaf disease prediction system on the smartphone using paas cloud</i>	Madhusudan G. Lanjewar, Kamini G. Panchbhai	3 kelas : <i>Tea leaf blight, Tea red leaf spot, Tea red scab</i> Data set didapatkan dari situs web mendeley Jumlah data set : 120	CNN (<i>Convolutional Neural Network</i>) dengan perbandingan model <i>NASNetMobile, Xception, dan ResNet50.</i>	CNN menggunakan 3 Layer konvolusi, dan variasi <i>pooling.</i> 8 Kelas data : <i>algal leaf, antharacnos, bird eye spot, brown blight, gray light, red leaf spot, white spot dan healthy</i>
2	<i>Diseases Classification for Tea Plant Using Concatenated Convolution Neural Network</i>	Dikdik Krisnandi, Hilman F. Pardede, R. Sandra Yuwana, Vicky Zilvan, Ana Heryana, Fani Fauziah, and Vitria Puspitasari Rahadi	4 kelas : Daun Sehat, penyakit melepuh, belalang daun, ulat looper Jumlah dataset : 4727	CNN (<i>Convolutional Neural Network</i>) dengan menggabungkan <i>GoogleNet, Xception, dan Inception-ResNet-v2</i>	CNN menggunakan 3 Layer konvolusi dan variasi <i>pooling.</i> 8 Kelas data : <i>algal leaf, antharacnos, bird eye spot, brown blight, gray light, red leaf spot, white spot dan healthy</i>
3	<i>Detection of Disease in Tea Leaves Using Convolution Neural Network</i>	Shyamtanu Bhowmik, Anjan Kumar Talukdar, and Kandarpa Kumar Sarma	3 kelas : <i>Healthy Tea Leaf, Black Rot of Tea, Rust of Tea</i> Dataset diambil sendiri Jumlah dataset : 2341	CNN (<i>Convolutional Neural Network</i>) dengan metode paling sederhana dengan menjaga kompleksitas komputasi minimum	CNN menggunakan 3 Layer konvolusi dan variasi <i>pooling.</i> 8 Kelas data : <i>algal leaf, antharacnos, bird eye spot, brown blight, gray light, red leaf spot, white spot dan healthy</i>

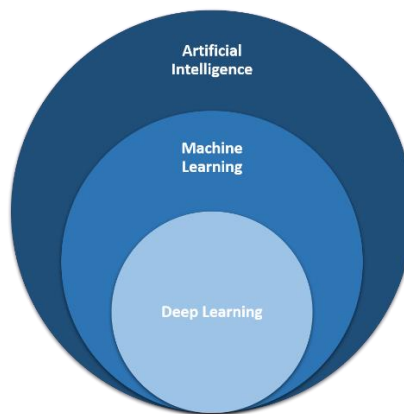
No	Judul	Nama Peneliti	Data	Metode	Perbedaan
4	Komparasi Algoritma <i>MobileNet</i> Dan <i>Nasnet Mobile</i> Pada Klasifikasi Penyakit Daun Teh	Nila Hardi	6 kelas : <i>algal spot, brown blight, gray blight, helopeltis, red spot dan healthy</i> Dataset diambil dari kaggle (Saikat Data, 2022)	CNN (<i>Convolutional Neural Network</i>) dengan arsitektur <i>mobileNet</i> dan <i>Nasnet Mobile</i>	CNN menggunakan 3 Layer konvolusi dan variasi <i>pooling</i> . 8 Kelas data : <i>algal leaf, antharacnos, bird eye spot, brown blight, gray light, red leaf spot, white spot</i> dan <i>healthy</i>
5	<i>Image Analysis and Detection of Tea Leaf Disease using Deep Learning</i>	S. Gayathri , D.C.Joy Winnie Wise, P. Baby Shamini, N. Muthukumaran	4 kelas : <i>blister blight, red scab, red leaf spot dan leaf blight</i> jumlah dataset : 500	CNN (<i>Convolutional Neural Network</i>) dengan arsitektur LeNet	CNN menggunakan 3 Layer konvolusi dan variasi <i>pooling</i> . 8 Kelas data : <i>algal leaf, antharacnos, bird eye spot, brown blight, gray light, red leaf spot, white spot</i> dan <i>healthy</i>
6	<i>Detection and Identifcation Of Tea Leaf Diseases Based On AX-RetinaNet</i>	Wenxia Bao, Tao Fan, Gensheng Hu, Dong Liang & Haidong Li	4 kelas : <i>tea algae leaf spot, tea bud blight, tea white scab dan tea leaf blight</i> jumlah dataset : 520	CNN (<i>Convolutional Neural Network</i>) berbasis AX- <i>RetinaNet</i>	CNN menggunakan 3 Layer konvolusi dan variasi <i>pooling</i> . 8 Kelas data : <i>algal leaf, antharacnos, bird eye spot, brown blight, gray light, red leaf spot, white spot</i> dan <i>healthy</i>

2.2 DASAR TEORI

2.2.1 Artificial Intelligence

Artificial Intelligence atau kecerdasan buatan adalah salah satu cabang pengetahuan didalam ilmu komputer yang sering disebut dengan nama AI.

Kecerdasan buatan dirancang dan dibuat untuk dimasukkan ke dalam mesin, seperti komputer, sehingga dapat melaksanakan pekerjaan dan bertindak seperti manusia. Hal ini dicapai dengan memberikan pengetahuan dan kemampuan penalaran yang baik kepada mesin tersebut. AI berupaya memahami dan memodelkan proses berpikir manusia, serta merancang mesin agar dapat meniru perilaku manusia. Para ahli sepakat bahwa AI merupakan sebuah proses berfikir manusia dan mempresentasikan proses tersebut kedalam mesin yang digunakan. Kemampuan AI untuk menyelesaikan masalah merupakan indikator utama dari kecerdasannya. Mesin yang dapat secara rutin menyelesaikan masalah dianggap sebagai mesin yang cerdas [15].



Gambar 2. 1 Bagian *artificial intelligence* [15].

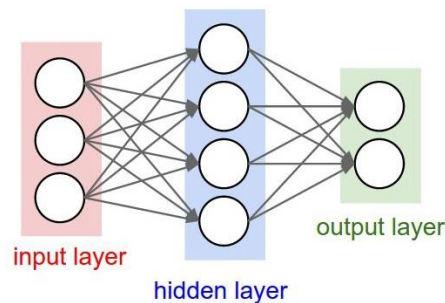
AI mencakup berbagai bidang ilmu, termasuk *machine learning*, yang di dalamnya terdapat *deep learning*, yang merupakan topik penelitian dalam konteks ini. Dari Gambar 2.1 dapat diketahui bahwa *artificial intelligence*, *machine learning*, dan *deep learning* saling berhubungan.

2.2.2 Deep Learning

Deep Learning merupakan subbidang dalam *machine learning* yang berbasis pada penggunaan Jaringan Saraf Tiruan (JST) yang dikembangkan untuk mengajarkan mesin atau komputer bagaimana melakukan tindakan seperti manusia. Pada *deep learning* mesin belajar dari contoh-contoh dan menerapkan pembelajaran tersebut dalam pengambilan keputusan atau tindakan. Dalam *deep*

learning, komputer mempelajari kemampuan untuk melakukan klasifikasi langsung dari teks, suara, atau gambar [16]

Deep Learning adalah sebuah teknik di dalam *Neural Network* yang menggunakan metode tertentu seperti *Restricted Boltzman Machine* (RBM) untuk mempercepat proses dalam pembelajaran pada *Neural Network* dengan banyak lapisan. Pada Gambar 2.2 lapisan yang terdapat pada *deep learning* terdiri dari tiga bagian yaitu *input layer*, *hidden layer* dan *output layer*. Pada *input layer* terdapat node-node yang masing-masing node menyimpan nilai masukan yang tidak berubah di fase latihan dan hanya berubah ketika menerima nilai masukan baru. Kemudian pada *hidden layer*, lapisan-lapisan dapat terbentuk untuk menemukan kombinasi algoritma yang optimal dalam upaya meminimalkan kesalahan (*error*) pada output. Pada output layer, fungsi aktivasi menghasilkan hasil perhitungan sistem yang didasarkan pada input yang diterima. *Output layer* bertanggung jawab untuk menampilkan hasil akhir dari proses tersebut [15].



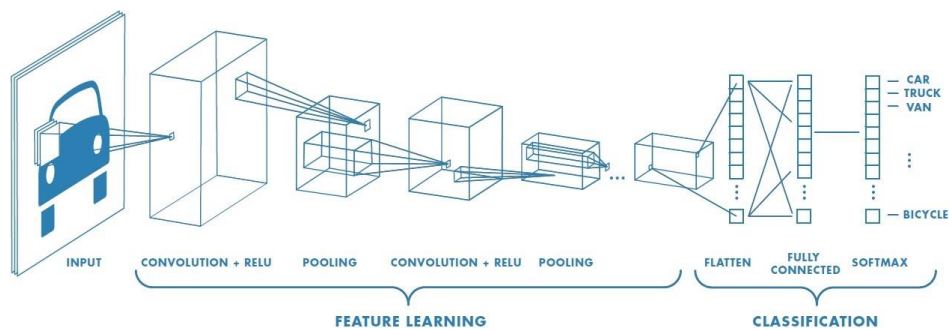
Gambar 2. 2 Layer pada *deep learning* [15].

2.2.3 *Convolutional Neural Network* (CNN)

Penemuan *Convolutional Neural Network* (CNN) pertama kali dilakukan oleh Hubel dan Wiesel saat melakukan penelitian mengenai *virtual cortex* pada kucing. Secara teknis, CNN merupakan suatu arsitektur yang dapat dilatih dan terdiri dari beberapa tahap, yaitu tahap masukan (*input*) dan tahap keluaran (*output*) [17]. CNN adalah salah satu arsitektur dalam *deep learning* yang sangat efektif dalam mengidentifikasi pola spasial. Dengan kemampuannya, CNN mampu mendeteksi dan mengenali objek yang terdapat pada sebuah gambar [18].

Pada CNN terdapat serangkaian proses yang melibatkan operasi konvolusi dan penggabungan lapisan untuk memproses data. Terdapat 4 tahapan yang akan

dilakukan pada CNN diantaranya input, tahap *ekstraksi fitur*, tahap klasifikasi dan *output*. Dengan tahapan ini, CNN dapat membentuk sistem yang mampu mengenali objek atau gambar dan bahkan melakukan aktivitas yang serupa dengan manusia. CNN akan melatih dan menguji setiap gambar masukan melalui serangkaian proses, yaitu *convolutional layer* yang diikuti oleh *pooling* untuk mengekstraksi fitur dari gambar masukan berturut-turut. Setelah operasi *pooling*, citra akan diubah menjadi *flatten* dan kemudian dimasukkan ke dalam proses *fully connected-layer* untuk melakukan tugas pengklasifikasian. Arsitektur CNN menggambarkan struktur dan urutan proses yang terjadi pada metode CNN [17].



Gambar 2. 3 Arsitektur dari CNN [17].

Dari Gambar 2.3 diketahui bahwa terdapat dua bagian lapisan arsitektur, yaitu *feature learning* dan *classification*. Bagian *feature learning* terdiri dari *convolutional layer* dan *pooling layer*. Pada *Classification* terdapat tiga jenis lapisan yang terdiri dari *flatten*, *fully connected* dan *softmax* [18]

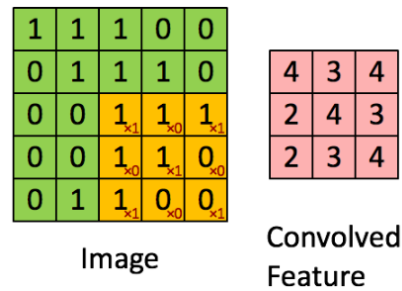
2.2.4 Feature Learning (Feature Extraction Layer)

Pada proses *feature learning*, terdapat lapisan-lapisan yang menerima *input* gambar secara langsung dan mengolahnya hingga menghasilkan keluaran berupa data *multidimensional array*. Lapisan tersebut terdiri dari lapisan konvolusi dan lapisan pooling.

A. Convolutional Layer

Convolution layer merupakan lapisan dasar yang berperan untuk melakukan ekstraksi fitur terhadap data input. Lapisan ini bertugas melakukan ekstraksi fitur dengan menggunakan operasi konvolusi yang melibatkan sekumpulan filter tiga

dimensi dan fungsi aktivasi tertentu untuk menghasilkan *feature map* [19]. Konvolusi berarti mengaplikasikan sebuah *kernel* yaitu kotak berwarna kuning pada citra seperti yang ditunjukkan pada Gambar 2.4. Seluruh citra yang akan dikonvolusi ditandai dengan kotak hijau. *Kernel* bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada gambar di sebelah kanannya [20].



Gambar 2. 4 Operasi konvolusi [20].

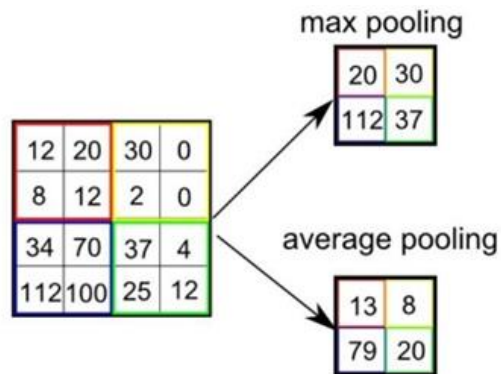
$$h = \frac{Input+2p-k}{s} + p \quad (2.1)$$

Pada persamaan 2.1 merupakan keluaran dari proses *convolution layer* atau disebut dengan *feature map*. Pada persamaan tersebut, h melambangkan *feature map* yang dihasilkan, p mewakili *padding*, k merupakan *kernel* yang digunakan, dan s menandakan *stride* yang digunakan.

B. Pooling Layer

Pooling layer adalah lapisan yang menggunakan fungsi dengan *feature map* untuk masukannya dan mengolahnya dengan berbagai jenis operasi statistik berdasar pada nilai piksel yang terdekat. Biasanya pada model CNN lapisan pooling disisipkan secara teratur setelah beberapa lapisan konvolusi. Dalam arsitektur model CNN lapisan pooling yang dimasukkan secara berturut-turut di antara lapisan konvolusi dapat secara progresif mengurangi ukuran volume output pada *feature map* sehingga dapat mengurangi jumlah perhitungan di jaringan dan parameter serta untuk pengendalian *overfitting*. Perhitungan *pooling* terbagi menjadi dua, yaitu *maximum pooling* dan *average pooling* seperti pada Gambar

2.5. *Maximum pooling* mengambil nilai maksimum dari *feature map*, sedangkan *average pooling* mengambil nilai rata-rata dari *feature map* [21].



Gambar 2. 5 Contoh operasi *pooling layer* [21].

Pada model CNN, lapisan *pooling* biasanya disisipkan secara teratur setelah adanya beberapa lapisan konvolusi. Dengan menempatkan lapisan *pooling* secara berurutan di antara lapisan konvolusi dalam arsitektur CNN, ukuran volume *output* pada *feature map* dapat secara bertahap dikurangi. Bentuk yang paling umum dari lapisan *pooling* adalah dengan menggunakan *kernel 2x2* yang diaplikasikan dengan langkah sebanyak dua dan kemudian beroperasi di setiap irisan dari *input*. Pada Gambar 2.5 menampilkan operasi dengan langkah 2 dan dengan ukuran *kernel 2x2*. Dari ukuran input 4x4, masing-masing empat angka pada input operasi mengambil nilai maksimalnya pada *maximum pooling* dan mengambil nilai rata-rata pada *average pooling* serta membentuk ukuran output baru menjadi 2x2. Proses tersebut dapat mengurangi jumlah parameter dan perhitungan dalam jaringan, serta untuk mengontrol *overfitting* [16].

C. ReLU (*Rectified Linear Unit*)

ReLU adalah lapisan aktivasi yang digunakan pada model CNN yang memiliki fungsi ini melakukan *thresholding* dengan nilai nol terhadap nilai piksel pada input citra. Pada persamaan 2.2 fungsi aktivasi ReLU bekerja dengan mengubah nilai negatif pada citra menjadi 0 [16]. Fungsi *ReLU* sebagai berikut :

$$f(x) = \max(0, x) \tag{2.2}$$

Keterangan :

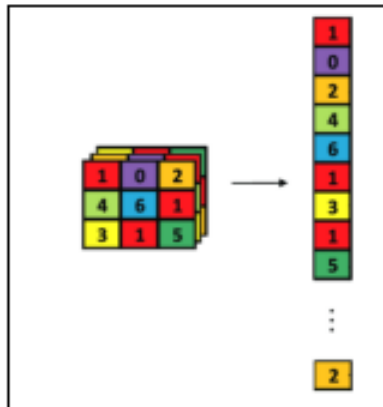
- $f(x)$ = Fungsi ReLu.
- x = Nilai batasan ReLu.
- $\max(0,x)$ = apabila $x \leq 0$ maka $x = 0$ dan apabila $x > 0$ maka $x = x$.

2.2.5 Classification Layer (Lapisan Klasifikasi)

Pada lapisan ini, terdapat neuron-neuron yang terhubung seluruhnya (*fully connected*) dengan lapisan lainnya. Lapisan ini menerima masukan (*input*) dari lapisan *output* pada bagian *feature learning* dan kemudian diproses melalui *flatten* dengan penambahan beberapa lapisan tersembunyi (*hidden layer*) pada *fully connected*. Hasil yang didapat adalah keluaran yang menunjukkan akurasi klasifikasi untuk setiap kelas [15].

A. Flatten

Flatten adalah operasi yang mengubah fitur (matriks) menjadi vektor satu dimensi. Dalam proses ini, *flatten* mengubah *feature map* dari lapisan sebelumnya menjadi vektor satu dimensi seperti pada Gambar 2.6 sehingga *feature map* tersebut dapat digunakan sebagai input bagi lapisan *fully connected* [22].



Gambar 2. 6 Ilustrasi *flattening* [7].

B. Fully Connected Layer

Fully Connected Layer atau sering disebut dengan *dense layer* adalah lapisan yang umumnya digunakan dalam implementasi *Multilayer Perceptron* (MLP) dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Sebelum dimasukkan ke dalam lapisan *fully*

connected, setiap *neuron* pada lapisan konvolusi perlu diubah menjadi data satu dimensi terlebih dahulu. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak dapat dipulihkan, *fully connected layer* hanya dapat diimplementasikan di akhir jaringan [20]. *Fully Connected Layer* memiliki peran penting dalam melakukan klasifikasi dengan membedakan antara kelas-kelas yang ada dalam data [23].

C. *Softmax*

Aktivasi *softmax* merupakan varian dari algoritma *logistic regression* yang dapat digunakan untuk mengklasifikasi lebih dari dua kelas. Algoritma Logistic Regression umumnya digunakan untuk tugas klasifikasi kelas biner. Pada aktivasi *softmax*, untuk persamaannya dapat dituliskan sebagai berikut :

$$F_j(Z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad (2.3)$$

Pada persamaan 2.3 notasi f_j menggambarkan hasil fungsi untuk setiap elemen ke- j dalam vektor *output* kelas. Argumen z adalah hipotesis yang diberikan oleh model pelatihan untuk mengklasifikasikan input dengan bantuan fungsi *softmax* [16].

2.2.6 *Epoch*

Epoch merupakan sebuah tahapan untuk melakukan pembelajaran dari seluruh set data *training*. Dalam jaringan saraf tiruan, proses pembelajaran iteratif dilakukan agar komputer dapat mengenali data yang diinputkan. Jika *epoch* yang tepat tidak dapat ditentukan secara pasti, model akan menentukan nilai *epoch* untuk mencapai tingkat akurasi yang optimal [24].

2.2.7 *Optimizer Adam*

Optimizer adalah metode yang bertujuan untuk menurunkan nilai *loss* sehingga menghasilkan akurasi terbaik dengan meng-update bobot. Ada beberapa optimizer yang umum digunakan seperti Adam. Adam (*Adaptive moment estimation*) merupakan pengganti untuk keturunan gradien stokastik dalam melatih model *Deep Learning*. Optimasi Adam menggabungkan sifat terbaik dari

algoritma RMSProp dan AdaGrad untuk menyediakan algoritma pengoptimalan yang mampu menangani gradien yang menyebar. Tahap pada *optimizer* Adam diulang sebanyak jumlah dataset yang diambil secara random sampai seluruh *epoch* telah selesai [15].

2.2.8 Confusion Matrix

Confusion Matrix memuat informasi tentang hasil dari model klasifikasi yang sebenarnya. Metode ini sering digunakan untuk menghitung akurasi, *recall*, dan *precision*. Terdapat empat istilah pada Tabel 2.2 sebagai representasi hasil proses klasifikasi yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN) [25].

Tabel 2. 2 Confusion matrix.

Kelas Aktual	Kelas Prediksi	
	Positive	Negative
Positive	TP (<i>True Positive</i>)	FN (<i>False Negative</i>)
Negative	FP (<i>False Positive</i>)	TN (<i>True Negative</i>)

Adapun pada persamaan 2.4 untuk menghitung akurasi dengan menggunakan *confusion matrix* adalah sebagai berikut :

$$\text{Akurasi} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \times 100 \% \quad (2.4)$$

TP : Jumlah data positif yang diklasifikasikan dengan benar.

FN : Jumlah data negatif namun diklasifikasikan salah.

TN : Jumlah data negatif yang diklasifikasikan dengan benar.

FP : Jumlah data positif namun diklasifikasikan salah.

2.2.9 Penyakit Daun Teh

Teh merupakan salah satu komoditas perkebunan yang memiliki peran penting pada perekonomian di Indonesia [26]. Hasil panen teh Indonesia setidaknya telah diekspor ke sekitar 78 negara dunia. Sektor perkebunan teh juga menjadi penyedia lapangan kerja terbesar dibandingkan dengan komoditas agrobisnis lainnya, mencapai sekitar dua juta orang di Indonesia [27]. Akan tetapi

teh tidak semuanya dapat dipanen dengan baik karena ada beberapa penyakit daun yang dapat menyerang pertumbuhan daun teh itu sendiri, sehingga mengakibatkan penurunan kualitas dari hasil panen daun teh yang berdampak pada ekspor teh. Adanya berbagai penyakit daun menyebabkan suatu perkebunan teh tidak dapat menghasilkan panen dengan maksimal. Penyakit daun teh diantaranya :

1. *Algal Leaf Spot*



Gambar 2. 7 Penyakit *algal leaf spot* [14].

Pada Gambar 2.7 penyakit *algal leaf spot* merupakan jenis penyakit daun teh yang ditandai oleh bintik bintik berwarna karat. Faktor penyebabnya adalah drainase tanah yang buruk, keseimbangan nutrisi yang tidak optimal, dan paparan suhu serta kelembaman yang relatif tinggi membuat tanaman teh rentan terhadap infeksi bercak daun alga [28].

2. *Anthracnose*



Gambar 2. 8 Penyakit *anthracnose* [14].

Pada Gambar 2.8 penyakit teh *anthracnose* adalah jenis penyakit teh yang disebabkan oleh jamur *Colletotrichum Corda*. Gejala awal yang ditimbulkan

ditandai dengan adanya lesi berair pada daun teh. Ketika penyakit berkembang, lesi menjadi lebih besar dan menjadi nekrotik, pada akhirnya menyebabkan kerugian [29].

3. Penyakit *Bird Eye Spot*



Gambar 2. 9 Penyakit *bird eye spot* [14].

Pada Gambar 2.9 penyakit teh bintik mata burung merupakan jenis penyakit teh yang sebagian terdapat di daerah Asia. Gejala yang ditimbulkan yaitu bintik bintik kecil seukuran lubang jarum pada daun teh. Hal tersebut dikarenakan cuaca yang mendung dan basah sangat mendukung terjadinya infeksi penyakit ini [28].

4. Penyakit *Brown Blight*



Gambar 2. 10 Penyakit *brown blight* [14].

Pada Gambar 2.10 penyakit teh busuk coklat atau *brown blight* adalah jenis penyakit tanaman teh yang disebabkan oleh jamur *C. scoparium* dan *G. cingulate*. Penyakit ini menyerang tanaman teh pada tahap pembibitan dan dapat menyebabkan kematian stek teh. Gejalanya akan timbul bercak-bercak coklat pada daun induknya, dimulai dari bagian ujung atau dari ketiak daun [30]

5. Penyakit *Gray Blight*



Gambar 2. 11 Penyakit *gray blight* [14].

Pada Gambar 2.11 penyakit teh hawar abu-abu atau *gray blight* adalah jenis penyakit tanaman teh yang disebabkan oleh jamur *pestalotia longiseta*. Gejala pada daun dimulai bercak kecil berwarna coklat yang kemudian melebar. Bercak tersebut memiliki pusat berwarna abu-abu dengan tepi berwarna coklat. Penyakit ini akan muncul pada tanaman yang mengalami kekurangan unsur hara, dipetik secara berlebihan, mengalami kekeringan, terpapar angin kencang, dan terkena sinar matahari yang intens.

6. Penyakit *Red Leaf Spot*



Gambar 2. 12 Penyakit *red leaf spot* [14].

Pada Gambar 2.12 penyakit bercak daun merah merupakan salah satu jenis penyakit pada tanaman teh. Penyakit ini dikenal sebagai penyakit akar boria teh. Gejala awalnya ditandai dengan daun yang menguning, kemudian layu, dan kemudian melekat pada batang selama beberapa hari setelah mati dengan daun atmosfer lebat [28].

7. Penyakit *White Leaf Spot*



Gambar 2. 13 Penyakit *white spot* [14].

Pada Gambar 2.13 penyakit bercak putih atau *white leaf spot* merupakan salah satu jenis penyakit tanaman teh yang disebabkan oleh infeksi jamur. Gejala awal terlihat bintik-bintik kecil tembus cahaya, kemudian bercak tersebut akan melebar dengan pusat yang tidak berwarna dan dibatasi oleh cincin berwarna putih. Dalam beberapa kasus, gejala bercak daun putih disebabkan oleh kekurangan unsur hara pada tanaman seperti *Calcium* (Ca), *Nitrogen* (N) dan Kalium (K). Penyakit ini terlihat seperti terbakar yang memperlihatkan ciri seperti ada bercak putih, daun menggulung dan permukaan daun menjadi kering [31].