BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini, penulis menggunakan beberapa tinjauan pustaka untuk mendukung penelitian yang sedang dilaksanakan. Dibawah ini adalah ringkasan dari tinjauan pustaka yang digunakan oleh penulis, yang dapat ditemukan pada Tabel 2.1.

Tabel 2.1 Daftar Literatur

No	Penulis	Tahun	Judul
1.	(Pramana, Setyati and Kristian, 2020)	2020	Model CNN LeNet Dalam Pengenalan Jenis Golongan Kendaraan Pada Jalan Tol
2.	(Fadli Gunardi, 2022).	2022	Implementasi Augmentasi Citra pada Suatu Dataset
3.	(Micheal, 2022)	2022	Klasifikasi Spesies Kupu Kupu Menggunakan Metode *Convolutional Neural Network*
4.	(Septianto, Setyati and Santoso, 2018)	2018	Model CNN LeNet dalam Rekognisi Angka Tahun pada Prasasti Peninggalan Kerajaan Majapahit
5.	(Rochmawati, 2021)	2021	Analisa *Learning Rate* dan *Batch Size* pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer Adam

2.1.1 Literatur 1

Oleh Anggay Luri Pramana, Endang Setyati, Yosi Kristian (2020) dari Program Studi Teknologi Informasi, Fakultas Sains, Institut Sains dan Teknologi Terpadu Surabaya dengan judul Model CNN LeNet Dalam Pengenalan Jenis Golongan Kendaraan Pada Jalan Tol. Penelitian yang dilakuakan bertujuan untuk mengidentifikasi jenis kendaraan menggunakan metode *Convolutional Neural Network* (CNN) *LeNet* dengan skenario pengujian 2 proses yaitu *prepocessing* dan training, testing. Dataset diambil menggunakan kamera handphone di gerbang tol dengan total dataset 500 data yang terdiri dari 5 kelas. Berdasarkan hasil pengujian dataset menggunakan epoch 25, 50, 75, dan 100, akurasi yang diperoleh terus meningkat mulai dari 82% sampai dengan 95% akurasi tertinggi. Pada hasil prediksi meningkat dari 80% sampai dengan 86%.

2.1.2 Literatur 2

Oleh Muhammad Fadli Gunardi (2022) dari Jurusan Teknik Informatika, Fakultas Teknik Institut Teknologi Bandung dengan judul Implementasi Augmentasi Citra pada Suatu Dataset. Penelitian yang dilakukan bertujuan untuk mengenali jenis *doodle* dengan menggunakan metode CNN dengan skenario pengujian menggunakan 5 objek gambar, setiap objek memiliki 30 gambar, dengan total dataset sebanyak 150 gambar. Pengujian ini menggunakan *max pooling* dan *average pooling* pada augmentasi data untuk pengenalan objek. Nilai akurasi sebesar 81% pada *max pooling* dan 67% pada *average pooling*.

2.1.3 Literatur 3

Oleh Micheal dan Ery Hartati (2022) dari Prodi Informatika, Ilmu Komputer Dan Rekayasa, Universitas MDP dengan judul Klasifikasi Spesies Kupu Kupu menggunakan metode *Convolutional Neural Network* (CNN). Penelitian yang di lakukan untuk mengklasifikasikan spesies kupu – kupu menggunakan metode CNN dengan arsitektur *LeNet* dan VGG-16. Dataset yang digunakan pada penelitian ini sebanyak 5455 citra dan dibagi menjadi 250 data tes, 250 data valid, dan 4955 data train. Data diaugmentasi sebanyak 8000 data train dan 800 data tes pada setiap kelas, sehingga mendapatkan hasil tingkat akurasi tertinggi sebesar 93% menggunakan VGG-16 sedangkan menggunakan LeNet sebesar 67%.

2.1.4 Literatur 4

Oleh Tri Septianto, Endang Setyati, Joan Santoso (2018) dari Sekolah Tinggi Teknik Surabaya dengan judul Model CNN LeNet dalam Rekognisi Angka Tahun pada Prasasti Peninggalan Kerajaan Majapahit. Penelitian yang dilakukan untuk mengenali objek angka tahun yang ada pada prasasti peninggalan kerajaan Majapahit menggunakan model LeNet yang mempunyai 10 kelas, dataset mempunyai 100 gambar pada masing – masing kelas, jadi total dataset sebesar 1000 citra. Pada model LeNet mendapatkan nilai akurasi 85,08% dengan menggunakan 10 epoch.

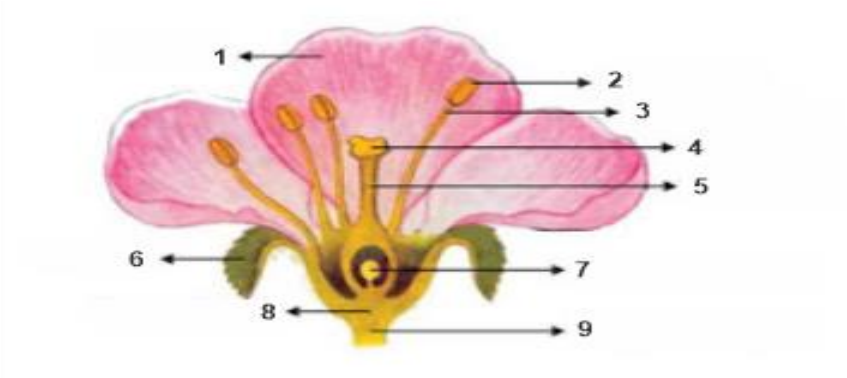
2.1.5 Literatur 5

Oleh Rochmawati (2021) dari Teknik Informatika Universitas Negeri Surabaya dengan judul Analisa *Learning Rate* dan *Batch Size* pada Klasifikasi

Covid Menggunakan *Deep Learning* dengan Optimizer Adam. Peneliti melakukan perbandingan pada beberapa *learning rate* dan *batch size* pada gambar medis penderita covid. Ada 6 *learning rate* dan 3 *batch size.* Pada penelitian 6 *learning rate* mendapatkan hasil optimal dengan nilai 0.0001 dan 0.00001. Sedangkan *batch size* yang paling bagus hasilnya dari tiga angka yang dibandingkan adalah batch size 5. Dataset yang digunakan pada penelitian ini dibagi menjadi 3, yaitu train, val, dan test. Terdapat 5186 data train, 520 test, 519 val.

2.2 Anggrek

Anggrek dengan nama latin *orchidaceae* yang memiliki bentuk dan ukuran bunga yang beragam. Anggrek memiliki struktur dasar tiga kelopak, dan tiga tajuk bunga. Anggrek termasuk bunga komoditas holtikultura yang memiliki nilai jual tinggi, sehingga banyak orang yang mengoleksi berbagai spesiesnya. Angrek memiliki kurang lebih 5.000 spesies di Indonesia, untuk membedakan macam spesies anggrek dapat dilihat dari warna, kelopak bunga, dan tekstur bunga. Namun secara umum spesies anggrek mempunyai kemiripan warna, kelopak bunga, dan teksturnya yang membuat kesulitan dalam mengidentifikasi spesies anggrek (Irawati, et al., 2021). Bagian – bagian bunga anggrek dapat dilihat pada **Gambar 2.1**.



Gambar 2.1 Bunga Anggrek (*orchidaceae*)

Sumber : (Anon., 2020)

Berikut bagian – bagian yang ada pada bunga anggrek :

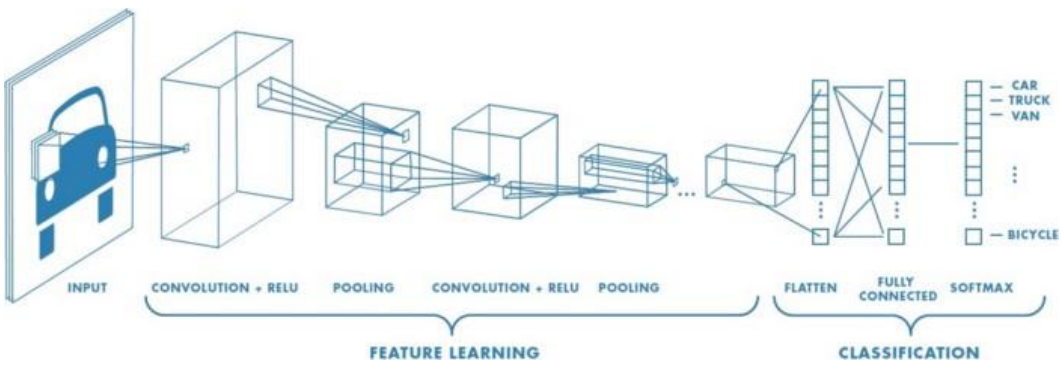
1. Mahkota
2. Kepala Sari
3. Benang Sari
4. Kepala Putik
5. Tangkai Putik
6. Kelopak
7. Bakal Biji
8. Dasar Bunga
9. Tangkai

2.3 Deep Learning

Deep Learning merupakan cabang dari *machine learning* menerapkan sistem jaringan saraf tiruan yang terinspirasi pada otak manusia. *Deep Learning* merupakan pengembangan dari *Multilayer Preceptron* (MPL) yang digunakan untuk mengolah data 2 dimensi seperti gambar dan suara. *Deep learning* menggunakan banyak data dan lapisan untuk membaca, memproses, menyimpan, dan mengklasifikasi objek. Metode *Deep Learning* mengandalkan CPU dan RAM pada proses komputasi, dan juga memanfaatkan GPU supaya proses komputasi berlangsung cepat (Batubara and Awangga, 2020).

2.4 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah suatu model jaringan syaraf yang berkembang dari *Multi Layer Perceptron* (MLP) dan terinspirasi dari sistem optik manusia. CNN adalah jenis model *Deep Neural Network* yang banyak digunakan oleh data citra karena kedalaman jaringan yang tinggi. CNN dikembangkan oleh Kunihiko Fukushima, seorang peneliti dari AT&T Bell Laboratories di Holmdel, New Jersey, USA. CNN adalah suatu arsitektur yang terdiri dari beberapa tahap uji yang dilatih untuk mengklasifikasi citra, mendeteksi objek, dan melakukan segmentasi, terdiri dari tahap masukan (*input layer*), keluaran (*output layer*), dan lapisan tersembunyi (*hidden layer*) (Almryad et al., 2020). CNN memiliki beberapa lapisan yang dapat dilihat pada **Gambar 2.2**.



Gambar 2.2 Model CNN

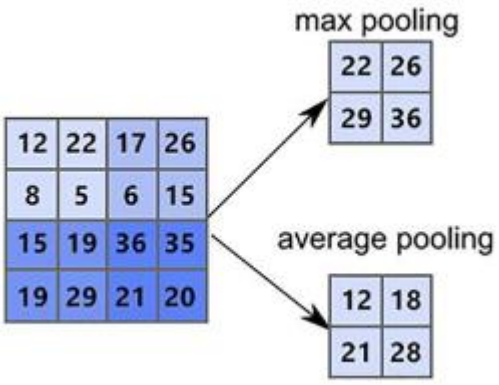
Sumber : (Almryad *et al.*, 2020)

2.4.1 *Convolutional Layers*

Convolutional layers merupakan lapisan utama pada blok CNN yang digunakan untuk proses citra yang masuk. Lapisan didalamnya terdiri dari filter - filter yang dipelajari secara acak dalam operasi konvolusi yang digunakan untuk ekstrasi fitur sebagai reprentasi fitur dari *input layer* (Pangestu, et al.., 2020).

2.4.2 *Pooling Layers*

Pooling layers atau *subsampling layer*s merupakan lapisan yang memiliki fungsi untuk mengurangi ukuran spasial dari fitur konvolusi yang dapat mengurangi sumber daya komputasi yang dibutuhkan untuk memproses data. Dengan adanya pengurangan dimensi melalui *feature map* dapat mempercepat komputasi, karena parameter yang diperbarui semakin sedikit (Pangestu, et al.., 2020). *Pooling layers* memiliki dua jenis tipe yaitu *max pooling* dan *avarage pooling*. Contoh *max pooling* dan *avarage pooling* dapat dilihat pada **Gambar 2.3**.



Gambar 2.3 *Max Pooling* dan *Average Pooling*

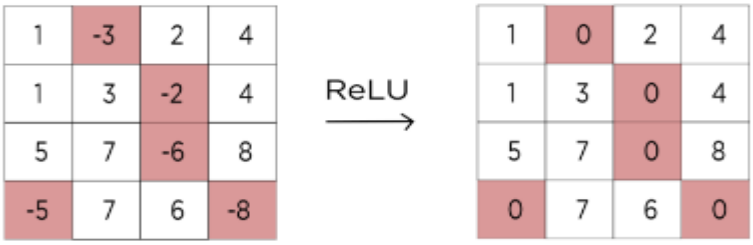
Sumber : (Pangestu, et al.., 2020)

2.4.3 *Fully Connected Layers*

Fully connected layers adalah lapisan yang digunakan dalam penerapan MLP untuk melakukan transformasi pada dimensi data agar dapat diklasifikasi secara linear. Dengan menggunakan komputasi perkalian matriks yang diikuti pada *bias offset*, pada operasi tersebut setiap neuron memiliki koneksi penuh ke semua aktivasi dalam lapisan sebelumnya (Pangestu, et al.., 2020).

2.4.4 Fungsi Aktivasi ReLU

Rectified linear units (ReLU) mempunyai fungsi untuk merubah atau menghilangkan nilai negatif pada citra. Perubahan nilai tersebut bertujuan untuk meningkatkan kualitas citra pada objek untuk meminimalisir kesalahan (Pangestu, et al.., 2020). Fungsi aktivasi ReLU untuk mengganti nilai negatif pada citra dengan nilai 0 dapat dilihat pada **Gambar 2.4**.

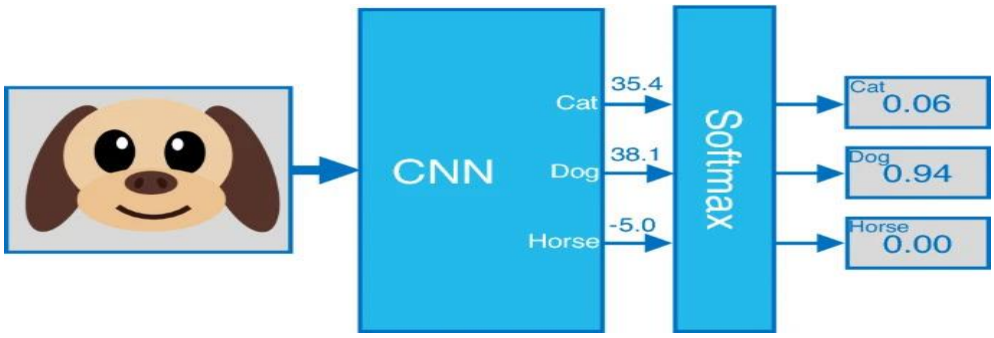


Gambar 2.4 Proses ReLU

Sumber : (Pangestu, et al.., 2020)

2.4.5 Fungsi Aktivasi *Softmax*

Softmax merupakan fungsi sebagai input vektor dari bilangan real K, dan menormalisasi menjadi probabilitas yang terdistribusi dari probabilitas K. Sebelum menggunakan *softmax*, beberapa komponen vektor bernilai negatif atau lebih besar dari 1 bahkan tidak berjumlah 1. Setelah menggunakan *softmax* setiap komponen berada dalam interval (0-1) dan komponen bertambah hingga 1, sehingga mereka dapat disebut probabilitas. Komponen input yang lebih besar akan sesuai dengan probabilitas yang lebih besar (Pangestu, et al.., 2020). Persamaan 1 pada fungsi aktivasi *softmax* sebagai berikut : $p(x) = \frac{e^x}{\sum_{k=1}^{k} e^x}$ fungsi aktivasi softmax dapat dilihat pada **Gambar 2.5.**

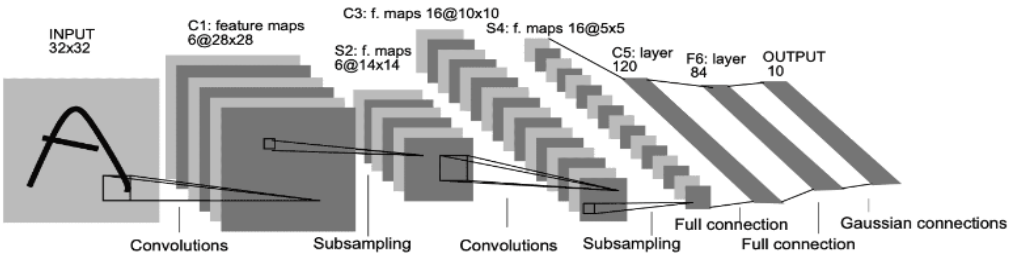


Gambar 2.5 Proses *Softmax*

Sumber : (Pangestu, et al.., 2020)

2.5 LeNet

LeNet adalah salah satu jaringan syaraf tiruan yang memiliki banyak lapisan berbasis *Convolutional Neural Network,* yang pertama kali dikenalkan oleh *Yann LeCun*. *LeNet* terbentuk dari pengembangan – pengembangan versi sebelumnya, sehingga *LeNet* mempunyai jumlah parameter yang banyak. Dengan banyaknya jumlah parameter yang dimiliki, membuat *LeNet* mampu untuk melakukan perhitungan matematis secara cepat (Fitriati, 2016). Arsitektur *LeNet* dapat dilihat pada **Gambar 2.6**.



Gambar 2.6 Arsitektur *LeNet*

Sumber : (Anon., 2022)

Input pada LeNet adalah gambar dengan ukuran 32x32x3 bersekala RGB melalui lapisan *convolutional* 6 *feature maps* dengan ukuran filter 5x5 dan menggunakan satu *stride*. 6 *feature maps* ini merupakan channel dari gambar yang sudah melakukan operasi konvolusi dengan ukuran 28x28x6. *Stride* digunakan untuk mengatur pergeseran suatu filter pada lapisan saat melewati *kernel size*. Lapisan kedua (S2) merupakan bagian *pooling layer* dengan filter

2x2, terdapat 6 *feature maps* dan dua *stride*. Lapisan ini mirip dengan lapisan sebelumnya menggunakan fungsi aktivasi *ReLU*.

Lapisan *convolutional* kedua memiliki 16 *feature maps* dengan ukuran filter 5x5 dan menggunakan aktivasi *ReLU* dan satu *stride* dan mendapatkan dimensi gambar dengan ukuran 10 x10x16. Lapisan keempat (S4) adalah lapisan *pooling layer* dengan *max pooling layer* atau *average pooling layer* dengan ukuran filter 2x2 dan fungsi aktivasi yang digunakan masih *ReLU*. Lapisan ini terdapat 400 *nodes* yang akan terhubung dengan gambar berdimensi 5x5x16.

Lapisan kelima (C5) merupakan lapisan *fully connected layer* dengan *feature maps* pada masing – masing ukuran menggunakan fungsi *ReLU*. Setiap 120 *nodes* pada lapisan ini terhubung ke 400 *nodes* yang ada pada lapisan keempat (S4). Lapisan keenam (F6) adalah lapisan *fully connected layer* dengan 84 *nodes* dan memiliki 10164 parameter latih. Lapisan ini menghubungkan hasil dari lapisan sebelumnya. Lapisan terakhir atau *output layer* adalah *fully connected layer* yang menggunakan fungsi aktivasi *softmax* dengan *size* berdasarkan hasil *output* gambar yang diklasifikasikan.

2.6 Prepocessing

Prepocessing merupakan sebuah proses untuk mengoptimalkan kualitas gambar sebelum gambar diolah, sehingga memudahkan kemampuan sistem dalam mengidentifikasi objek (Ibrahim et al., 2022).

2.7 Augmentasi Data

Augmentasi data merupakan proses modifikasi suatu citra sehingga komputer mendeteksi citra yang dimodifikasi adalah citra berbeda, akan tetapi manusia dapat mengetahui bahwa citra yang dimodifikasi adalah citra yang sama. *Augmentasi data* dapat meningkatkan variasi sampel data pelatihan serta mencegah terjadinya *overfitting*. *Overfitting* adalah kondisi model yang mempelajari objek pada proses pelatihan dengan baik, akan tetapi memberikan

hasil yang buruk pada pada proses pengujian (Perez and Wang, 2018). *Augmentasi data* dapat meningkatkan ukuran dan kualitas data pelatihan sehingga model yang dihasilkan lebih baik dari sebelumnya. Teknik ini dapat menghasilkan data berupa gambar yang telah diproses menggunakan parameter tertentu seperti : *rotation*, *zoom*, *width_shift*, *height_shift*, *shear*, *horizontal_flip*, dan *vertical_filp* (Perez and Wang, 2018).

Model yang digunakan telah menggunakan data baru dengan sedikit modifikasi untuk diuji, dengan otomatis komputer akan mempelajari dengan baik ketika variasi data ditamabahkan pada saat proses pelatihan (Shorten and Khoshgoftaar, 2019). Augmentasi data yang sering digunakan yaitu : *rotation*, *color space transformation*, dan *flipping*.

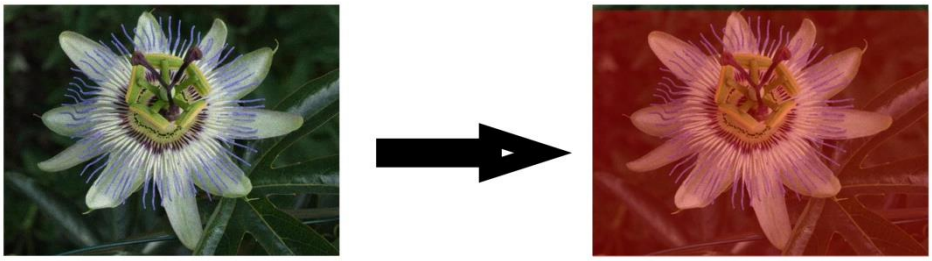
2.7.1 *Rotation*

Rotation (rotasi) merupakan teknik pada *augmentasi data* yang digunakan untuk memutar gambar. Gambar dapat diputar dengan nilai 1º - 359º ke arah kanan maupun kiri (Shorten and Khoshgoftaar, 2019). Hasil dari teknik *rotation* dapat dilihat pada **Gamabar 2.7**.

Gambar 2.7 Teknik *Rotation*

2.7.2 *Color Space Transformation*

Color space transformation merupakan teknik perubahan komposisi warna yang digunakan pada *augmentasi data* untuk menambah variasi data dengan komposisi warna yang berbeda dari masing - masing gambar (Shorten and Khoshgoftaar, 2019). Hasil dari teknik *color space transformation* dapat dilihat pada **Gambar 2.8**.



Gambar 2.8 Teknik *Color Space Transformation*

2.7.3 *Flipping*

Flipping merupakan teknik pada augmentasi data yang digunakan untuk membalik gambar secara *vertical* maupun *horizontal*. Teknik ini mudah diterapkan serta mampu meningkatkan kinerja model klasifikasi pada saat proses pengujian (Shorten and Khoshgoftaar, 2019). Hasil dari teknik *flipping* dapat dilihat pada **Gambar 2.9**.

Gambar 2.9 Teknik *Flipping*

2.8 Confusion Matrix

Confusion matrix merupakan salah satu metode yang digunakan untuk menghitung tingkat akurasi gambar. *Confusion matriks* sangat berguna untuk menganalisis seberapa baik *classifier* mengenali sebuah data yang termasuk ke dalam kategori data uji benar dan kategori data uji salah. Tabel confusion matrix dapat dilihat pada Tabel 2.2.

Tabel 2.2 *Confusion Matrix*

Actual Class	**Predicted Class**		**Total**
	Yes	**No**	
Yes	*TP*	*FN*	*P*
No	*FP*	*TN*	*N*
Total	*P′*	*N′*	*P + N*

Keterangan :

- TP (*True Positive*), jumlah data dengan nilai sebenarnya positif dan nilai prediksi positif.
- FP (*False Positive*), jumlah data dengan nilai yang sebenarnya negative dan nilai prediksi positif.
- FN (*False Negative*), jumlah data dengan nilai sebenarnya positif dan nilai prediksi negative.
- TN (*True Negative*), jumlah data dengan nilai sebenarnya negatif dan nilai prediksi negatif.

Nilai dari TP dan TN adalah informasi ketika classifier benar dalam melakukan klasifikasi data, sedangkan nilai FP dan FN adalah informasi ketika classifier salah dalam melakukan klasifikasi data. Berdasarkan penelitian sebelumnya model yang digunakan untuk mengevaluasi kinerja pada parameter *confusion matrix* adalah *accuracy*, *recall*, *precision,* dan *f1 score* (Yüzkat *et al.*, 2021).

2.8.1 *Accuracy*

Accuracy (akurasi) adalah matrik yang umum digunakan dalam melakukan evaluasi klasifikasi. Cara kerja akurasi, dengan menghitung nilai probabilitas berdasarkan nilai yang benar dari label kelas. Akurasi memiliki persamaan 2, seperti berikut (Yüzkat *et al.*, 2021).

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad \text{.....} (2)$$

2.8.2 *Recall*

Recall atau *sensitivity* adalah keakuratan dari nilai positif yang ada. *Recall* memiliki persamaan 3, seperti berikut.

$$recall = \frac{TP}{TP+FN} \times 100\% \quad \text{.....(3)}$$

2.8.3 *Precision*

Precision adalah nilai kebenaran dari prediksi yang dilakukan melalui *classifier* dengan label kelas yang sudah ada. *Precision* memiliki persamaan 4, sebagai berikut.

$$precision = \frac{TP}{TP+FP} \times 100\% \quad \text{.....(4)}$$

2.8.4 *F1 Score*

F1 score adalah nilai rata – rata harmonik dari recall dan precision. F1 score dinyatakan dengan persamaan 5, sebagai berikut.

$$f1\ score = \frac{TP}{TP+\frac{1}{2}(FP+FN)} \times 100\% \quad \text{.....(5)}$$

2.9 Hyperparameter

Hyperparameter adalah konfigurasi yang berada diluar model. Tugas *hyperparameter* yaitu membantu dalam menemukan parameter model yang tidak bergantung pada data *training* (Rochmawati, 2021). *Hyperparameter* yang sering digunakan yaitu *epoch*, *batch size*, *learning rate* dan *optimizer*.

2.9.1 *Epoch*

Epoch merupakan *hyperparameter* yang menentukan berapa kali proses akan dilakuakannya masa *training* dalam *neural network* (Rochmawati, 2021).

2.9.2 *Batch Size*

Batch size adalah jumlah *training sample* yang digunakan dalam satu *iteration*. *Batch size* juga digunakan dalam proses *training* untuk menentukan jumlah contoh data *training* dan merupakan salah satu *hyperparameter* terpenting (Rochmawati, 2021).

2.9.3 *Learning Rate*

Learning rate merupakan salah satu *hyperparameter training* yang digunakan untuk menghitung nilai koreksi bobot pada masa proses *training* (Rochmawati, 2021).

2.9.4 *Optimizer Adam*

Optimizer adam merupakan algoritma *stokastik* yang digunakan untuk memperbarui bobot secara iteratif yang di dasarkan oleh data training. *Optimizer adam* ini merupakan kombinasi antara *RMSprop* dan *Stochastic*, sangat cocok diterapkan pada permasalahan data dengan *gradient* yang menyebar (Rochmawati, 2021).

2.10 Loss Function

Loss function digunakan untuk membandingkan nilai hasil dari prediksi dengan nilai yang sebenarnya (Batubara and Awangga, 2020). *Cross entrophy* termasuk salah satu *loss function* yang digunakan dalam klasifikasi pada kasus *multi class classification*. *Cross entrophy* dinyatakan dalam persamaan 6 sebagai berikut.

$$\text{error} = \sum_i E_t \quad \text{.....(6)}$$

Dimana E_t dapat dilihat pada persamaan 7.

$$E_t = -\sum_i y_t \times \log o_t \quad \text{.....(7)}$$

Pada persamaan 7 terdapat beberapa variable yaitu, y_t yang termasuk kelas sebenarnya pada timestep ke-t, sedangkan o_t adalah kelas prediksi pada timestep ke t, dan E_t adalah eror dari output o_t.