

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada penyusunan penelitian ini, dilakukan beberapa studi literatur yang relevan dengan topik penelitian. Proses pengkajian terhadap penelitian terdahulu dapat menjadi referensi penting bagi penelitian ini. Berikut adalah beberapa penelitian terdahulu yang relevan:

Penelitian pertama yang berjudul "*Self-supervised Transformer for Deepfake Detection*" yang dilakukan pada tahun 2022. Penelitian ini dilakukan dengan tujuan mengatasi masalah pemalsuan wajah, dalam berkembang pesatnya teknologi manipulasi wajah yang diberdayakan oleh model generative yang mendalam membuat deepfake lebih realistis dan mudah menipu pengamat. Data yang digunakan adalah dataset VoxCeleb2AV Speech untuk pra-pelatihan, total data berisi 2.800.000 video klip pidato. Hasil dari penelitian ini adalah deteksi pemalsuan wajah dengan menggunakan model *Transformer*, dimana model menunjukkan kinerja yang sebanding dengan yang tercanggih dalam banyak kasus [1].

Penelitian selanjutnya berjudul "*Fake Generated Painting Detection Via Frequency Analysis*" yang dilakukan pada tahun 2020. Penelitian ini dilakukan dengan tujuan untuk mendeteksi pemalsuan lukisan digital yang dapat dihasilkan dengan berbagai algoritma transfer gaya. Data yang digunakan dalam penelitian ini adalah data lukisan asli dan lukisan palsu, kumpulan data pelatihan untuk lukisan AI berjumlah 4992 data yang dihasilkan melalui tiga trans-gaya canggih, dan kumpulan data pelatihan lukisan asli berjumlah 4128 data yang berasal dari lima belas seniman terkenal. Hasil dari penelitian ini adalah deteksi lukisan palsu atau asli menggunakan metode yang diberi nama FGPD-FA dengan menggunakan fitur domain fourier, HOG, gaussian, dan diklasifikasikan dengan SVM, dimana hasil penelitian ini berkinerja baik dan dapat digunakan untuk mendeteksi lukisan palsu [7].

Penelitian selanjutnya berjudul "*Deep Convolutional Pooling Transformer For Deepfake Detection*" yang dilakukan pada tahun 2022. Penelitian ini dilakukan dengan tujuan mendeteksi gambar wajah palsu untuk mengatasi masalah keamanan dan privasi yang tersebar luas di internet. Data yang digunakan adalah data video asli dan palsu,

yang diperoleh dari Face Forensics (FF++) yang berisi 1.000 video asli dari YouTube dan 1.000 video palsu disintesis dari 1.000 video asli. Hasil dari penelitian ini adalah deteksi wajah palsu dengan menggunakan model *transformer* konvolusional, dimana penelitian ini menyimpulkan bahwa model yang lebih dalam tidak selalu menghasilkan kinerja yang lebih baik dibandingkan dengan model deteksi yang lebih dangkal [4].

Penelitian selanjutnya berjudul “*Temporal-Spatial Time Series Self-Attention 2D & 3D Human Motion Forecasting*” pada tahun 2022. Penelitian ini dilakukan dengan tujuan meningkatkan kewaspadaan terhadap benda bergerak di lingkungan, peneliti berfokus pada peramalan gerakan manusia berdasarkan data 2D dan 3D menyajikan metode *Self-Attention Temporal-Spatial Time Series* untuk peramalan gerak manusia. Data yang digunakan adalah kumpulan data *Human3.6M*, *3DPW*, dan *AMASS* berdasarkan protokol evaluasi standar. Hasil dari penelitian ini metode yang digunakan bekerja sangat baik dalam data *ground truth* dan estimasi pose 2D dibandingkan dengan metode deret waktu lainnya. Metode kami belum mengungguli penelitian sebelumnya dalam data masukan 3D. Namun, berdasarkan penilaian kuantitatif dan kualitatif, pendekatan kami menunjukkan kinerja yang sangat baik dalam memprediksi pergerakan manusia untuk tujuan jangka pendek dan panjang [8].

Tabel 2.1 Ringkasan Penelitian Sebelumnya

No. Referensi	Metode	Hasil	Kesamaan	Perbedaan
[1]	<i>Transformer</i>	Deteksi pemalsuan wajah dengan menggunakan model <i>transformer</i> , dengan model kinerja yang sebanding dengan yang tercanggih dalam banyak kasus.	Penerapan metode <i>Transformer</i> .	Penelitian sebelumnya mendeteksi pemalsuan wajah, sedangkan pada penelitian ini mendeteksi lukisan potret palsu dari model <i>Generative</i> .
[7]	FGPD-FA	Deteksi lukisan palsu atau asli yang memberikan kinerja yang baik dengan menggunakan metode FGPD-FA dengan fitur domain fourier, HOG, Gaussian, dan diklasifikasikan dengan SVM.	Mendeteksi lukisan dari hasil model <i>generative</i> .	Penelitian sebelumnya mendeteksi lukisan hasil model <i>Generative</i> dengan gaya <i>Van Gogh</i> , sedangkan penelitian ini mendeteksi lukisan dengan gaya potret dengan subjek manusia.
[4]	<i>Deep Convolutional Pooling Transfomer</i>	Deteksi wajah palsu yang dapat menyimpulkan bahwa model lebih dalam tidak selalu menghasilkan kinerja yang lebih baik.	Penerapan metode CNN dan <i>Transformer</i> .	Penelitian sebelumnya mendeteksi pemalsuan wajah, sedangkan pada penelitian ini mendeteksi lukisan potret palsu dari model <i>generative</i> .

[8]	<i>Self- Attention Temporal- Spatial Time Series</i>	Peramalan gerak manusia dengan metode yang digunakan bekerja sangat baik dalam data <i>ground truth</i> dan estimasi pose 2D dibandingkan dengan metode deret waktu lainnya. Metode kami belum mengungguli penelitian sebelumnya dalam data masukan 3D. Namun, berdasarkan penilaian kuantitatif dan kualitatif, pendekatan kami menunjukkan kinerja yang sangat baik dalam memprediksi pergerakan manusia untuk tujuan jangka pendek dan panjang.	Penerapan metode <i>Transformer</i> .	Penelitian sebelumnya <i>Transformer</i> digunakan untuk peramalan gerak manusia, sedangkan pada penelitian ini digunakan untuk mendeteksi lukisan potret palsu hasil model <i>Generative</i> .
-------	--	--	---------------------------------------	---

Berdasarkan tabel 2.1 diatas, terdapat kemiripan dan perbedaan dari penelitian yang akan dilakukan penulis pada penelitian selanjutnya. Kemiripan dari penelitian terdahulu dengan penelitian yang akan dilakukan selanjutnya yaitu pada metode yang digunakan dan merupakan model mengenai solusi dalam klasifikasi, serta menggunakan tingkat akurasi untuk hasil pengujiannya. Kemudian, untuk perbedaan antara penelitian terdahulu dengan penelitian yang akan dilakukan peneliti selanjutnya adalah objek penelitian, waktu penelitian, dan studi kasus penelitian.

2.2 Landasan Teori

2.2.1 Seni Lukis

Menurut Leo Tolstoy, seni adalah aktivitas manusia yang dilakukan secara sadar, melalui tanda-tanda lahiriah tertentu, menularkan emosi yang dialaminya kepada orang lain agar orang lain dapat bergairah dan mengalami emosi serupa. Seni rupa diekspresikan melalui sarana visual atau intuitif seperti titik, garis, bentuk, warna, tekstur, volume dan ruang. Seni rupa dapat didefinisikan sebagai “ekspresi estetika melalui titik, garis, bentuk, warna, tekstur, volume dan ruang” [9].

Dalam penggunaan sehari-hari, istilah “seni rupa” dan “desain” seringkali dianggap sebagai dua kata yang berbeda, namun dalam konteks penjajarannya, istilah “seni rupa” dipahami sebagai seni murni, dimana istilah “desain” dikaitkan dengan seni terapan. Seni rupa memiliki unsur fisik dan unsur non fisik. Dimana unsur fisik tersebut berupa bentuk, warna, tekstur, ruang, dan struktur. Sedangkan unsur immaterial adalah isi yang ada dalam karya seni yang kemudian dimaknai oleh pengamatnya. Dalam komposisinya, seni rupa mempunyai prinsip-prinsip seperti kesatuan, keseimbangan, proporsi, ritme, fokus dan kontras. Karya seni memiliki beragam jenisnya tergantung pada bentuknya, yaitu karya seni dua dimensi, karya seni tiga dimensi, dan karya seni empat dimensi [9].

Karya seni rupa dua dimensi digambarkan dengan hanya dapat diapresiasi dalam satu arah dan hanya memiliki dimensi panjang dan lebar [9]. Seni lukis termasuk kedalam karya seni rupa dua dimensi yang digambarkan melalui sapuan cat atau warna dan berasal dari imajinasi seniman yang dituangkan dalam bidang dua dimensi. Hasil seni yang paling humanistik berupa potret seseorang atau kelompok merupakan seni lukis potret. Dalam seni lukis potret, seorang seniman melampaui tugas sederhana menggambarkan ekspresi fisik dari subjeknya, mereka cenderung menyajikan aspek psikologis (konsep-konsep psikologis yang muncul dari keahlian seorang seniman) dalam karyanya [2].

2.2.2 *Artificial Intelligent*

Pada perkembangan awalnya, komputer digunakan untuk memproses data mentah yang kemudian disajikan melalui antarmuka agar dapat dipahami oleh manusia, terutama dalam konteks pengambilan keputusan. Namun, seiring perkembangan berikutnya, kecerdasan buatan mendorong komputer tidak hanya memproses data, tetapi juga memiliki kemampuan pengetahuan. Kecerdasan buatan pada dasarnya merupakan bagian dari ilmu komputer yang berusaha mengadopsi kemampuan berpikir manusia ke dalam perangkat komputer [10].

Kecerdasan buatan terdiri dari tiga komponen utama, yaitu struktur data yang tersusun secara sistematis, algoritma yang disesuaikan dengan berbagai masalah untuk menghasilkan pengetahuan, dan penggunaan bahasa pemrograman pada tahap teknis akhir untuk menciptakan perangkat lunak sesuai dengan kebutuhan. Dalam konteks penerapan kecerdasan buatan, komponen utamanya adalah basis pengetahuan (*knowledge base*) dan mesin inferensi (*inference engine*) [10].

Bidang Kecerdasan Buatan memiliki beberapa cabang ilmu yang berkembang pesat, termasuk Sistem Pakar, Penalaran Komputer, Pembelajaran Mesin, Jaringan Saraf Tiruan, Pemrosesan Bahasa Alami, Pengenalan Pola, penalaran komputer, Robotika, dan Sistem Pendukung Keputusan. Setiap cabang ini memiliki peran penting dalam memajukan kecerdasan buatan dan mengaplikasikannya dalam berbagai konteks [10].

a. Sistem Pakar (*Expert System*)

Sistem Pakar merupakan perangkat lunak yang berfokus pada pengetahuan dan memiliki kemampuan untuk menjalankan tugas-tugas yang sebelumnya hanya dapat dilakukan oleh manusia dengan keahlian khusus.

b. Penalaran Komputer (*Computer Reasoning*)

Penalaran komputer yang berlandaskan pada kasus adalah sebuah pendekatan yang berbeda dari metode-metode kecerdasan buatan lainnya. Pendekatan ini terbukti efektif dan sering digunakan dalam menyelesaikan masalah yang serupa dengan cara manusia berpikir.

- c. **Pembelajaran Mesin (*Machine Learning*)**

Pembelajaran mesin adalah teknologi yang memungkinkan sistem untuk mengakumulasi pengetahuan dan pengalaman langsung dari data. Penerapan pembelajaran mesin memiliki potensi aplikasi yang luas, termasuk di bidang kesehatan, pendidikan, transportasi, dan berbagai bidang lainnya.
- d. **Jaringan Saraf Tiruan (*Artificial Neural Network*)**

Jaringan saraf tiruan adalah model matematis yang berupaya mensimulasikan struktur dan fungsi jaringan saraf biologi, menciptakan representasi yang meniru proses pembelajaran otak manusia.
- e. **Pemrosesan Bahasa Alami (*Natural Language Processing*)**

Pemrosesan bahasa alami bisa dibagi menjadi dua aspek utama, yaitu pemahaman bahasa alami dan generasi bahasa alami. Pemahaman bahasa alami melibatkan kemampuan untuk mengerti teks, sementara generasi bahasa alami melibatkan kemampuan untuk membuat teks. Pemrosesan bahasa alami memainkan peran penting dalam pengembangan model dan proses yang dapat menerima informasi dalam bentuk suara, teks, atau keduanya sebagai input, dan kemudian mengolahnya menggunakan algoritma komputer.
- f. **Pengenalan Pola (*Pattern Recognition*)**

Pengenalan pola diterapkan untuk memberikan solusi terhadap beragam masalah, mulai dari mengenali ucapan, mengidentifikasi wajah, hingga mengklasifikasikan karakter tulisan tangan dan melakukan diagnosis medis.
- g. **Indra Komputer (*Computer Vision*)**

Tujuan dari computer vision adalah mengembangkan sistem yang mampu secara mandiri menjalankan berbagai tugas.
- h. **Robotika**

Robotika adalah disiplin ilmu yang luas dan tak terbatas yang menggabungkan unsur-unsur dari berbagai bidang, termasuk kecerdasan buatan, otomatisasi, dan pengendalian jarak jauh. Terdapat empat

karakteristik inti yang mendefinisikan robotika, yakni kemampuan untuk mendeteksi lingkungan melalui sensor, kemampuan pengendalian yang terhubung dengan kecerdasan, sistem peralatan mekanik yang berfungsi sebagai aksi dan pergerakan, serta sumber daya daya yang memungkinkan operasi robot.

i. Sistem Pendukung Keputusan (*Decision Support System*)

Sistem Pendukung Keputusan merupakan sistem informasi yang menggunakan komputer untuk membantu dalam memilih solusi terbaik dari berbagai alternatif yang ada untuk suatu masalah. Komponen kunci dari sistem ini adalah model sistem pendukung keputusan.

2.2.3 *Machine Learning*

Machine learning adalah sub bidang ilmu komputer yang telah berevolusi dari studi pengenalan pola dan teori pembelajaran komputasi dalam konteks kecerdasan buatan. Dalam *machine learning*, algoritma dirancang untuk dapat memahami data dan membuat prediksi berdasarkan pola yang ada dalam data tersebut. Ilmu ini erat kaitannya dengan statistik komputasi, yang juga berfokus pada pembuatan prediksi berdasarkan data. *Machine learning* terbagi menjadi tiga kategori utama, tergantung pada tipe pembelajaran yang digunakan [11]:

- a. Pembelajaran yang diawasi, di mana komputer diberikan contoh *input* dan *output* yang diinginkan, serta tujuannya adalah mempelajari aturan umum yang menghubungkan *input* dan *output*.
- b. Pembelajaran Tanpa Pengawasan, dimana komputer diberikan data tanpa label, serta tujuannya adalah menemukan struktur dalam data tersebut.
- c. Pembelajaran Penguatan, di mana komputer berinteraksi dengan lingkungan dinamis dan harus mencapai tujuan tertentu tanpa petunjuk eksplisit.

Terdapat berbagai jenis algoritma *machine learning*, termasuk regresi, klasifikasi, dan pengelompokan (*clustering*). Dalam konteks ini, regresi adalah algoritma yang memahami hubungan antara variabel dependen dan variabel

independen. Klasifikasi adalah pendekatan yang mengklasifikasikan data berdasarkan pelatihan yang ada. *Clustering* bertujuan untuk mengelompokkan data yang serupa tanpa memerlukan label. Ada berbagai jenis algoritma *Machine Learning* yang digunakan, diantaranya Regresi Linear, Klasifikasi Biner, K-Means *Clustering*, *Algoritma Decision Tree*, dan Algoritma Naive Bayes [11].

2.2.4 *Deep Learning*

Deep learning (DL) adalah sebuah cabang penting dalam bidang kecerdasan buatan, telah menjadi salah satu bidang yang paling menjanjikan baik untuk penelitian maupun industri. Meskipun DL adalah topik yang sangat modern dan sudah digunakan oleh beberapa perusahaan teknologi terbesar, seperti Google dalam berbagai aplikasi seperti pengenalan suara dan gambar. DL melibatkan penggunaan arsitektur jaringan saraf tiruan (ANN) yang dalam, yang memiliki banyak lapisan. Namun, tak ada batasan pasti mengenai berapa banyak lapisan yang dapat dikategorikan sebagai "dalam". Salah satu karakteristik utama DL adalah penggunaan dataset yang besar, yang semakin bertambah seiring dengan pertumbuhan volume data yang diperoleh [12].

DL merupakan sub kategori dari ML di mana model atau jaringan belajar langsung dari dataset, dan kata "deep" merujuk pada jumlah lapisan yang digunakan dalam jaringan tersebut. DL sangat cocok untuk aplikasi modern seperti pengenalan wajah, penerjemahan teks, pengenalan suara, dan sistem bantuan pengemudi canggih, seperti pengenalan jalur dan rambu lalu lintas. Ada beberapa alasan mengapa DL dianggap canggih [12]:

- a. Penggunaan alat dan teknik canggih telah secara signifikan meningkatkan akurasi algoritma DL. Dibandingkan dengan ML pada masalah yang sama dengan dataset yang sama, DL telah terbukti mengurangi tingkat kesalahan secara drastis.
- b. DL mampu menangani dataset yang sangat besar, yang memungkinkan pelatihan jaringan yang lebih efisien. Dalam dunia saat ini, data sangat melimpah.

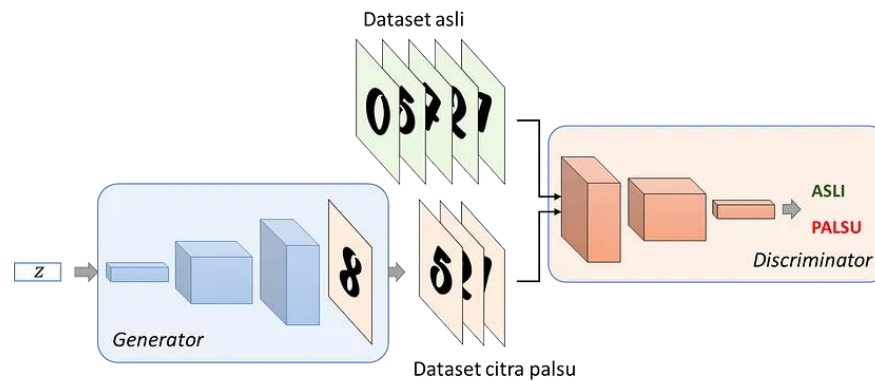
Penggunaan GPU berkinerja tinggi telah mengakselerasi pelatihan jaringan DL, yang sebelumnya memakan waktu berminggu-minggu, menjadi hanya beberapa jam. GPU yang canggih ini tersedia di berbagai platform cloud yang mendukung DL.

2.2.5 Klasifikasi Gambar

Klasifikasi gambar merupakan suatu proses di mana gambar-gambar dikelompokkan ke dalam kategori atau label tertentu menggunakan algoritma pembelajaran mesin. Algoritma klasifikasi gambar adalah algoritma pembelajaran yang bersifat terawasi dan mendukung klasifikasi multi-label. Proses ini melibatkan penggunaan gambar sebagai *input*, dan satu atau lebih label diberikan sebagai output untuk setiap gambar. Algoritma ini menggunakan jaringan saraf konvolusional yang dapat dilatih dari awal atau menggunakan pembelajaran transfer saat terdapat keterbatasan dalam jumlah besar gambar pelatihan yang tersedia. Kesimpulannya, Klasifikasi gambar adalah langkah di mana gambar-gambar dikelompokkan ke dalam kategori-kategori atau label-label tertentu dengan bantuan algoritma pembelajaran mesin, seperti jaringan saraf konvolusi (CNN). Proses klasifikasi gambar ini bermanfaat dalam berbagai konteks, termasuk analisis forensik gambar, pengenalan wajah, dan pembuatan model-model klasifikasi gambar [13].

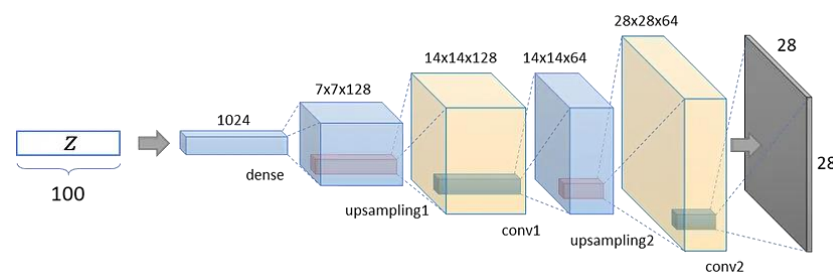
2.2.6 *Generative Adversarial Network (GAN)*

Algoritma GAN adalah sebuah kerangka kerja untuk model generatif yang memanfaatkan kombinasi dari dua jenis model. Komponen utamanya terdiri dari model generatif (G) yang bertanggung jawab untuk mempelajari distribusi data, dan model diskriminatif (D) yang mengestimasi probabilitas bahwa sampel berasal dari data pelatihan daripada model generatif [14].



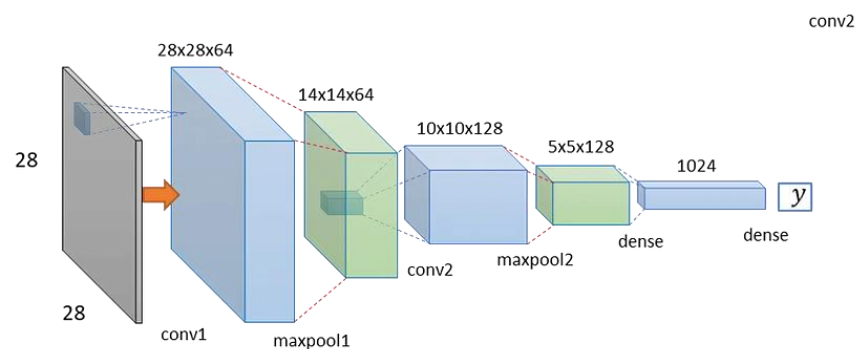
Gambar 2.1 Arsitektur GAN [15]

Dalam tahap awal pelatihan, generator menerima *input* berupa array yang dihasilkan secara acak dengan distribusi khusus yang disebut sebagai latent space (z). Array ini kemudian diubah menjadi citra melalui proses konvolusi [14].



Gambar 2.2 Arsitektur Generator [15]

Pada tahap berikutnya, diskriminator menerima *input* yang dihasilkan oleh generator dan mengeluarkan output berupa nilai biner yang menunjukkan apakah citra masukan berasal dari dataset asli (1) atau merupakan citra palsu (0) [14].

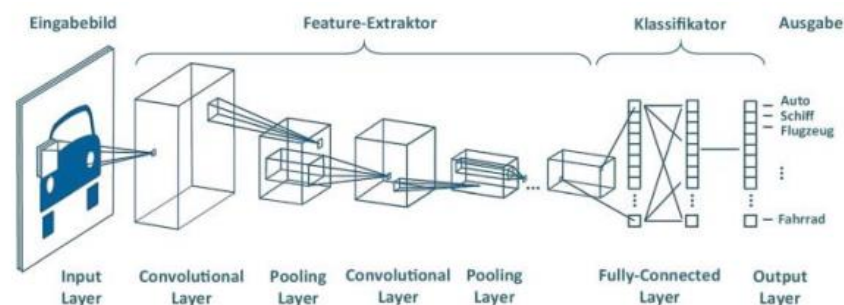


Gambar 2.3 Arsitektur Diskrikminator [15]

Generator dikatakan berhasil ketika mampu menghasilkan gambar atau data yang sangat realistis sehingga sulit dibedakan dari data asli oleh manusia maupun oleh discriminator [14].

2.2.7 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah salah satu arsitektur jaringan saraf *Deep Learning* yang dirancang untuk mengolah data dua dimensi. Biasanya digunakan untuk mempelajari dan mendeteksi *feature* pada sebuah gambar [5]. CNN terdiri dari berbagai lapisan, termasuk lapisan input, lapisan konvolusi, lapisan penggabungan, dan lapisan terhubung penuh. Lapisan konvolusi bertugas menerapkan filter pada gambar masukan untuk mengekstrak fitur, lapisan penggabungan digunakan untuk mengurangi ukuran gambar guna mengurangi perhitungan, dan lapisan terhubung penuh digunakan untuk membuat prediksi akhir. CNN adalah jenis jaringan saraf yang canggih untuk memahami gambar, CNN belajar melalui latihan dan mencoba memahami piksel gambar. Model ini dapat memotong gambar menjadi potongan-potongan kecil dan memeriksa setiap potongan seperti mengamati piksel. CNN sangat baik dalam mengidentifikasi apa yang ada dalam gambar, seperti membedakan kucing dan anjing, dan digunakan dalam banyak hal, seperti mengenali wajah, menemukan objek dalam gambar, atau bahkan mendiagnosis penyakit dari gambar medis [16]. Berikut ini menunjukkan tahapan pada model CNN dalam memproses suatu *input* gambar.



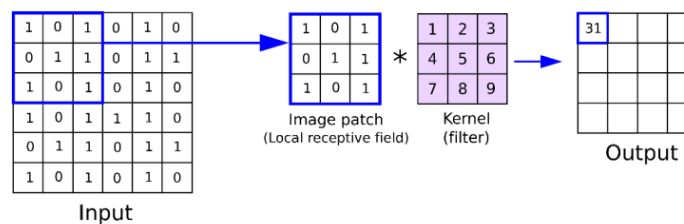
Gambar 2.4 Arsitektur *Convolutional Neural Network* [17]

Berdasarkan Gambar 2.4, arsitektur CNN terbagi menjadi dua bagian besar yaitu, *feature extraction* dan *classification*. *Feature extraction* terdiri atas

convolutional layer dan *pooling layer*, sedangkan *classification* terdiri atas *flatten*, *dropout regularization*, *fully connected layer*, dan *softmax* [18]:

1. Convolution layer

Convolution layer adalah elemen inti dalam arsitektur jaringan saraf konvolusi (CNN) yang bertanggung jawab untuk menjalankan operasi konvolusi pada output dari layer sebelumnya. Operasi konvolusi ini melibatkan penerapan fungsi pada output dari fungsi lain secara berulang. Penggunaan yang paling umum adalah untuk mengekstraksi fitur dari gambar, di mana filter digunakan untuk menggeser gambar, memindai beberapa piksel sekaligus, dan menghasilkan peta fitur yang mengidentifikasi setiap fitur yang ditemukan. Ini dapat dijelaskan sebagai berikut [18]:



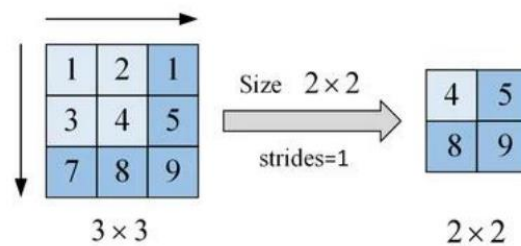
Gambar 2.5 Ilustrasi *Feature Map*

Pada gambar 2.5 di atas, *convolution layer* terdiri dari neuron yang membentuk filter dengan panjang, lebar, dan ketebalan. Pada gambar berukuran 5x5x3 (panjang 5 piksel, lebar 5 piksel, dan ketebalan 3 sesuai dengan saluran gambar). Filter ini kemudian digeser ke seluruh bagian gambar. Selama perpindahan ini, operasi "dot" dilakukan antara input dan nilai filter sehingga menghasilkan peta aktivasi output atau peta fitur.

2. Pooling Layer

Pooling layer adalah langkah setelah *convolutional layer*, yang bertujuan untuk mengurangi kompleksitas perhitungan tanpa menghilangkan fitur-fitur utama. Layer ini terdiri dari sebuah penyaring dengan langkah tertentu yang secara bergantian digerakkan ke seluruh area peta fitur. Dalam layer penyaringan, terdapat dua jenis penyaringan yang umum digunakan, yaitu penyaringan rata-rata (*average pooling*) dan penyaringan maksimum (*max*

pooling). Dalam *average pooling*, nilai rata-rata matriks input dihitung berdasarkan cakupan filter-nya, sedangkan *max pooling* mengambil nilai terbesar dari matriks input berdasarkan cakupan filter dari lapisan *max pooling*. *Pooling layer* ini menggunakan filter berukuran 2×2 yang diterapkan dengan langkah sebanyak dua dan beroperasi pada setiap sebagian inputnya. Operasi *max pooling* diilustrasikan sebagai berikut [18]:



Gambar 2.6 Operasi *Max Pooling*

Gambar 2.6 menggambarkan proses *max pooling* menggunakan filter berukuran 2×2 dengan input yang berukuran 3×3 . Pada setiap operasi, tiga angka pada input diambil nilai maksimalnya dengan hasilnya adalah matriks output dengan ukuran baru 2×2 .

3. *Dropout*

Dropout adalah metode regulasi dalam jaringan saraf yang bertujuan untuk memilih beberapa neuron secara acak dan mengabaikannya selama proses pelatihan. Ini berarti kontribusi dari neuron yang dinonaktifkan akan dihentikan sementara dalam jaringan, dan pembaharuan bobot juga tidak diterapkan pada neuron tersebut selama proses *backpropagation* [18].

4. *Flatten Layer*

Flatten layer terletak di antara *convolutional layer* dan *fully connected layer*. Tugas utama dari layer ini adalah mengonversi data berdimensi dua menjadi vektor berdimensi satu. Penggunaan layer ini diperlukan karena gambar dan output dari jaringan konvolusional umumnya berdimensi dua, sementara *fully connected layer* hanya menerima input satu dimensi [18].

5. *Fully Connected Layer*

Fully connected layer adalah suatu lapisan yang menghubungkan setiap neuron dalam satu lapisan ke lapisan berikutnya. Lapisan ini umumnya

digunakan dalam MLP (*Multi-Layer Perceptron*) dengan tujuan untuk melakukan transformasi dimensi data sehingga data dapat diklasifikasikan secara linear. Perbedaan antara layer ini dengan *convolutional layer* adalah bahwa neuron dalam lapisan konvolusi hanya terhubung dengan area tertentu dalam input, sedangkan *fully connected layer* memiliki koneksi antara neuron secara keseluruhan. Meskipun demikian, fungsi dari kedua jenis lapisan ini serupa karena keduanya melakukan operasi perkalian “dot” [18].

6. Fungsi Aktivasi

Fungsi aktivasi digunakan dalam jaringan saraf untuk mengatur kapan neuron harus diaktifkan atau tidak [17]:

a. ReLU

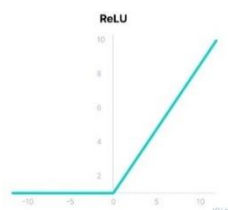
ReLU merupakan fungsi linear yang telah dimodifikasi sehingga akan menghasilkan keluaran sesuai dengan input positif, tetapi akan mengembalikan keluaran nol jika menerima input negatif. Fungsi ini juga berperan dalam mengatasi isu gradien yang hilang dan memberikan efisiensi komputasi. Persamaan matematis dari fungsi aktivasi ReLU adalah sebagai berikut [19]:

$$f(x) = \max(0, x) \quad (2.1)$$

Apabila x lebih besar dari 0, dan apabila x kurang dari atau sama dengan 0.

$$f(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (2.2)$$

Grafik Fungsi aktivasi ReLU dapat diilustrasikan sebagai berikut:



Gambar 2.7 Grafik Fungsi Aktivasi: ReLU

a. Linear

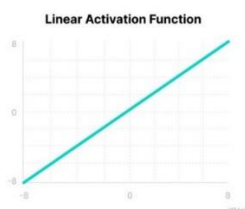
Fungsi aktivasi linear juga dikenal sebagai 'tanpa aktivasi' atau 'fungsi identitas', memiliki karakteristik di mana fungsinya proporsional dengan input yang diberikan. Fungsi ini tidak melakukan modifikasi pada jumlah bobot input dan mengeluarkan nilai sebagaimana adanya. Terdapat dua kekurangan pada fungsi ini, yaitu [19]:

- Tidak memungkinkan penggunaan propagasi mundur karena turunan fungsinya konstan dan tidak terkait dengan input x .
- Penggunaan fungsi aktivasi linear akan mengakibatkan semua lapisan dalam jaringan saraf menjadi satu, karena lapisan terakhir akan tetap menjadi fungsi linier dari lapisan pertama. Dengan kata lain, fungsi aktivasi linear secara efektif mengubah jaringan saraf menjadi satu lapisan.

Persamaan matematis dari fungsi aktivasi linear adalah sebagai berikut:

$$f(x) = x \quad (2.3)$$

Sedangkan Grafik Fungsi aktivasi Linear dapat diilustrasikan sebagai berikut:



Gambar 2.8 Grafik Fungsi Aktivasi Linear

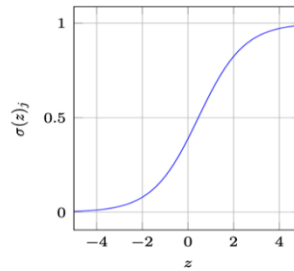
b. *Softmax*

Softmax memiliki fungsi untuk mengubah vektor bilangan riil menjadi vektor yang berisi probabilitas untuk setiap kategori. Komponen-komponen dalam vektor output berada dalam interval $(0, 1)$, dan total probabilitas selalu sama dengan 1. Penggunaan umum dari softmax adalah sebagai fungsi aktivasi di lapisan terakhir jaringan saraf dalam tugas klasifikasi, terutama dalam kasus tugas multi-kelas.

Fungsi aktivasi *Softmax* memiliki persamaan matematis sebagai berikut [19]:

$$\text{softmax}(y_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2.4)$$

Fungsi *softmax* dapat diilustrasikan sebagai berikut:



Gambar 2.9 Grafik Fungsi Aktivasi: *Softmax* [20]

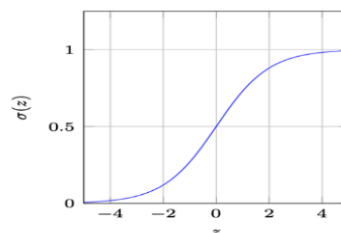
c. *Sigmoid*

Fungsi *sigmoid* menghasilkan nilai di dalam rentang (0, 1), yang bermanfaat dalam melakukan perhitungan yang harus diinterpretasikan sebagai probabilitas. Selain itu, ini menormalisasi data dan mendukung pembentukan output probabilistik serta pembuatan fungsi kerugian yang turun dari model kemungkinan maksimum. Fungsi aktivasi ini umumnya digunakan pada lapisan terakhir dari jaringan klasifikasi, terutama ketika menangani tugas biner. Sigmoid lebih efisien karena hanya melibatkan 1 neuron pada layer terakhirnya, sementara Softmax tetap menggunakan 2 neuron pada kasus klasifikasi biner.

Fungsi ini memiliki persamaan matematis sebagai berikut [19]:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.5)$$

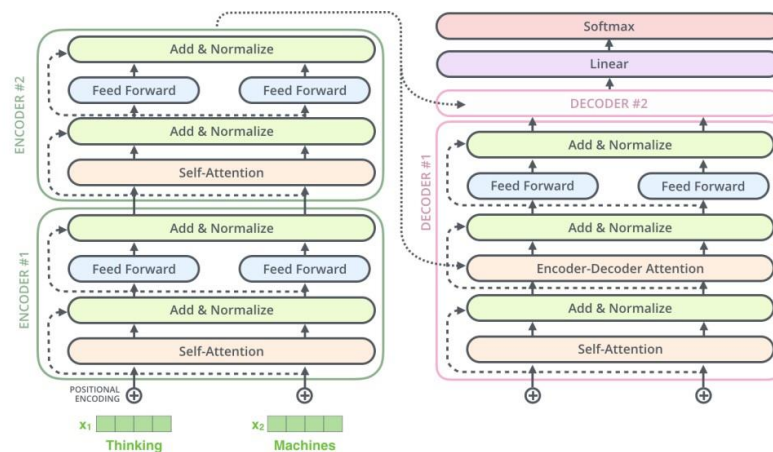
Fungsi *sigmoid* ini diilustrasikan sebagai berikut:



Gambar 2.10 Grafik Fungsi Aktivasi: *Sigmoid* [20]

2.2.8 Transformer

Transformer adalah model *Deep Learning* yang diperkenalkan pada tahun 2017 pada paper *Attention is all you need* [21]. *Transformer* digunakan secara luas dalam bidang pemrosesan bahasa alami (*natural language processing*). Arsitektur *Transformer* dirancang khusus untuk menangani data sekuensial, seperti teks, dan telah berhasil diterapkan dalam berbagai tugas NLP, termasuk penerjemah mesin dan ringkasan teks [22]. Salah satu keunggulan utama *Transformer* adalah kemampuannya untuk memahami konteks dalam teks dengan menggunakan mekanisme yang disebut perhatian atau *self-attention*, yang memungkinkan pengolahan teks yang lebih efisien dan kontekstual [23].

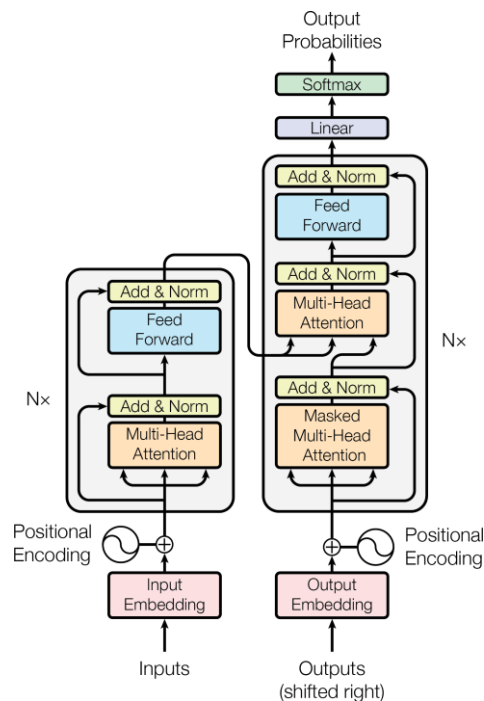


Gambar 2.11 Arsitektur *Encoder Decoder* pada *Transformer*

Encoder adalah bagian yang terdiri dari 6 lapisan atau tingkatan pemrosesan [22]. Setiap lapisan encoder memiliki dua bagian yang disebut sub-lapisan [24]. Sub-lapisan pertama adalah seperti otak yang dapat memperhatikan informasi dalam beberapa cara (*multi-head self-attention mechanism*). Sub-lapisan kedua adalah seperti pabrik yang memproses data dengan lebih rinci (*Position-wise fully connected feed-forward network*). Setiap lapisan encoder juga memiliki lapisan normalisasi yang membuat hasilnya menjadi lebih stabil. Hasil dari setiap lapisan encoder adalah

gabungan dari hasil sub-lapisan pertama dan kedua, setelah melalui lapisan normalisasi [22].

Decoder juga terdiri dari 6 lapisan yang bekerja serupa dengan *encoder*. Setiap lapisan *decoder* memiliki tiga sub-lapisan [22]. Dua sub-lapisan pertama mirip dengan yang ada di *encoder* [24]. Sub-lapisan ketiga *decoder* berfokus pada informasi yang diperoleh dari *encoder*. Decoder harus memahami dan memproses hasil yang telah dihasilkan oleh *encoder*. Seperti di *encoder*, setiap lapisan *decoder* memiliki lapisan normalisasi untuk menjaga hasilnya tetap konsisten. Sub-lapisan ketiga menggunakan masking untuk memastikan bahwa prediksi yang dibuat hanya bergantung pada informasi yang diketahui sebelumnya dan tidak melibatkan informasi di masa depan [22].



Gambar 2.12 Arsitektur *Transformer*

1. Attention

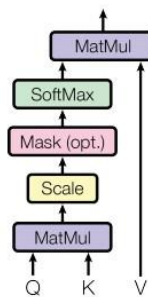
Fungsi attention dapat dideskripsikan sebagai pemetaan query dan sekumpulan pasangan *key-value* ke *output*, di mana *query*, *key*, *value*, dan *output* semuanya adalah vektor. Outputnya dihitung dari total bobot dari *value*, di mana bobot yang ditetapkan ke setiap nilai dihitung oleh fungsi kompatibilitas dari *query* dengan *key* yang sesuai [22].

a. *Scaled Dot-Product Attention*

Scaled dot-product attention terdiri dari *query* dan *key* dimensi d_k , dan *value* dimensi d_v . untuk menghitung *scaled dot-product*, *query* dikalikan dengan semua *key*, dan dibagi dengan $\sqrt{d_k}$, dan terapkan fungsi softmax untuk mendapatkan bobot pada *value* [22].

$$\text{Attention}(Q, K) = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \quad (2.6)$$

Scaled Dot-Product Attention

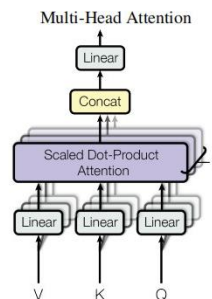


Gambar 2.13 Arsitektur *Scaled Dot-Product Attention*

b. *Multi-Head Attention*

Multi-head attention merupakan kumpulan fungsi *scaled dot-product attention*. *Multi-head attention* memungkinkan model untuk bersama-sama memperhatikan informasi dari representasi yang berbeda diposisi berbeda [22].

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h). W_o \quad (2.7)$$



Gambar 2.14 Arsitektur *Multi-Head Attention*

c. *Position-wise Feed-Forward Network*

Setiap layer encoder dan decoder terdapat *fully connected feed-forward network* yang terdiri dari 2 linear transformations dengan aktivasi ReLU [22].

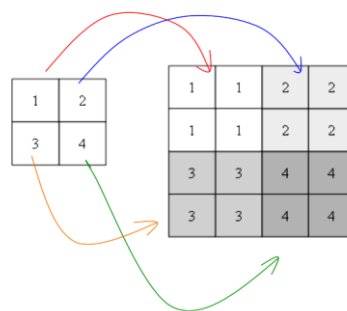
$$\text{FFN}(x) = \max(0, xW_1 + b_1) W_2 + b_2 \quad (2.8)$$

2.2.9 Preprocessing

Preprocessing dapat didefinisikan sebagai segala operasi yang memungkinkan untuk memperbaiki, menganalisa, atau mengubah suatu gambar, termasuk menghilangkan *noise* atau bagian gambar yang tidak diperlukan. Sebelum pengenalan karakter dilakukan, *preprocessing* dilakukan. Untuk mengetahui seberapa efektif proses pengenalan citra, tahap ini sangat penting. *Preprocessing* yang digunakan, diantaranya [25]:

2.2.9.1 Resizing

Resizing atau penskalaan adalah tahap dalam pengolahan citra yang melibatkan perubahan ukuran citra. Proses ini bertujuan untuk mengurangi ukuran citra, yang pada gilirannya membantu dalam mempercepat proses pengolahan citra oleh komputer dan menghemat ruang penyimpanan di memori sementara (*cache*) [25].



Gambar 2.15 *Resizing* (memperbesar citra) [26]

Resizing melibatkan perubahan resolusi atau dimensi horizontal dan vertikal citra. Proses ini dapat mencakup pembesaran, perubahan tinggi, perubahan lebar, atau kombinasi dari perubahan tersebut. Ketika kedua dimensi diubah, sebagian besar aplikasi pengolahan citra akan menjaga aspek rasio citra, menggunakan dimensi yang lebih besar sebagai

dasar dan mempertahankan aspek rasio citra asli. Ini berguna untuk menjaga kualitas citra saat mengubah ukuran. Dengan demikian, *resizing* dapat berarti memperbesar atau memperkecil citra, tergantung pada kebutuhan pengolahan [25].

Resizing pada posisi kolom dapat dilihat pada persamaan berikut [26]:

$$m = \frac{pb \times pp}{pa} \quad (2.9)$$

Dengan keterangan,

- m = posisi kolom piksel pada matriks baru
- pb = panjang matriks baru
- pp = posisi kolom piksel pada matriks lama
- pa = panjang matriks lama

Resizing pada posisi baris dapat dilihat pada rumus berikut:

$$n = \frac{lb \times pp}{la} \quad (2.10)$$

Dengan keterangan,

- n = posisi baris piksel pada matriks baru
- lb = lebar matriks baru
- pp = posisi baris piksel pada matriks lama
- la = lebar matriks lama

2.2.9.2 Normalization

Normalisasi adalah suatu metode pengolahan data yang bertujuan mengubah data asal menjadi format yang lebih cocok untuk analisis dan pemodelan. Proses normalisasi ini terutama berfokus pada pengaturan skala data, dengan tujuan menghindari perbedaan besar dalam rentang data awal. Dengan melakukan normalisasi, semua data diberikan bobot yang seimbang, sehingga perbedaan rentang data awal yang besar tidak mendominasi data dengan rentang yang lebih kecil [27].

[[50,	200,	150],
[100,	75,	25],
[0,	255,	175]]

a

[[0.196,	0.784,	0.588],
0.392,	0.294,	0.098],
0,	1,	0.686]]

b

Gambar 2.16 (a) nilai *pixel* sebelum normalisasi, (b) nilai *pixel* setelah normalisasi

$$x_{normal} = \frac{x - \min(X)}{\max(X) - \min(X)} \quad (2.11)$$

Dengan keterangan:

- x_{normal} = nilai yang dinormalisasi
- x = nilai awal
- $\min(X)$ = nilai minimum dari seluruh data
- $\max(X)$ = nilai maksimum dari seluruh data

2.2.10 Evaluasi Matriks

Evaluasi Matriks merujuk pada parameter atau tanda yang digunakan untuk mengukur performa model *machine learning*. Matriks ini berguna untuk mengevaluasi sejauh mana model *machine learning* dapat memprediksi target atau mengklasifikasikan data, serta membandingkan performa berbagai model dalam berbagai situasi. Penggunaan metriks evaluasi ini penting karena [28].

- a. Matriks membantu dalam menilai seberapa baik suatu model machine learning berkinerja, apakah dalam hal prediksi target atau klasifikasi data.
- b. Matriks digunakan sebagai alat pembanding antara berbagai model machine learning, membantu pemilihan model yang paling optimal.
- c. Matriks mendukung dalam mengidentifikasi area-area yang perlu perbaikan dalam model dan memberikan panduan untuk meningkatkan hasil yang dicapai.
- d. Matriks memungkinkan pengambilan keputusan berdasarkan data (*data-driven*) dan memastikan bahwa model beroperasi sesuai harapan.

Dengan kata lain, matriks evaluasi merupakan komponen penting dalam evaluasi dan pemilihan model *machine learning* serta membantu dalam meningkatkan kinerja dan hasil model tersebut. Berikut merupakan metrik untuk mengukur performa pada model *machine learning*:

2.2.10.1 Confusion Matrix

Confusion matrix adalah sebuah teknik yang digunakan dalam data mining dan *machine learning* untuk mengukur sejauh mana performa model dalam memprediksi label data. Model ini digambarkan dalam bentuk tabel yang mencatat berapa kali model memberikan prediksi yang benar dan berapa kali model salah. Model ini menghasilkan *output* yang dibagi menjadi dua kategori, yaitu positif atau negatif dan ya atau tidak. Masing-masing tabel *confusion matrix* terdiri dari empat sel yang mewakili kombinasi unik dari prediksi dan nilai aktual. Tabel *confusion matrix* dapat diperhatikan pada gambar 2.17 berikut [29]:

		Predicted value	
		P	N
True value	P	TP	FN
	N	FP	TN

Gambar 2.17 Tabel *Confusion Matrix* [30]

Dengan keterangan,

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.

2.2.10.2 Akurasi

Akurasi didapatkan dengan membagi jumlah data bernilai positif yang diprediksi positif dan data bernilai negatif yang diprediksi negatif dengan total data dalam dataset. Untuk menghitung akurasi dapat dirumuskan sebagai berikut [31]:

$$\text{Accuracy} = \frac{TP + FN}{TP + FP + FN + TN} \quad (2.12)$$

2.2.10.3 Recall

Recall didapatkan dari peluang kasus yang diprediksi positif dalam kategori positif.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.13)$$

2.2.10.4 Precision

Precision didapatkan dari kemungkinan kasus yang diprediksi positif yang juga termasuk kasus kategori positif.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.14)$$

2.2.10.5 F-1 Score

F-1 score juga dikenal sebagai F-Measure, F-score didapatkan dari hasil recall dan precision antara kategori hasil prediksi dan kategori sebenarnya.

$$F1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (2.15)$$

2.2.11 Cross-Entropy Loss

Loss function berperan untuk dapat mengukur kerugian yang terkait dengan kemungkinan yang dihasilkan model. Model dengan banyak kelas, diperlukan suatu metode untuk mengukur perbedaan antara probabilitas prediksi model dan probabilitas sebenarnya. Selama proses pelatihan, berbagai algoritma digunakan untuk menyesuaikan parameter model guna meminimalkan perbedaan. Konsep *Cross-Entropy* adalah dengan meminimalkan nilai negatif dari dataset [32]. Secara matematis, *Cross-Entropy Loss* didefinisikan [32]:

$$D(s, p) = - \sum_j p_j \log s_j \quad (2.16)$$

Dengan keterangan,

- s_j = nilai hasil prediksi
- p_j = nilai sesungguhnya
- j = Jumlah target kelas

2.2.12 Binary Cross-Entropy Loss

Binary Cross-Entropy (BCE) adalah *loss function* yang sering digunakan untuk model klasifikasi dengan dua kategori. BCE merupakan

varian dari *Cross-Entropy* (CE) yang disesuaikan untuk situasi di mana model hanya perlu membedakan antara dua kelas. Sementara *Cross-Entropy* dapat mengakomodasi klasifikasi dengan banyak kelas, BCE secara khusus dirancang untuk *binary classification* [14]. Secara matematis, *Binary Cross-Entropy* didefinisikan [14]:

$$\text{loss}(x, y) = - \sum x \log(y) \quad (2.17)$$

Dengan keterangan,

x = nilai aktual
 y = nilai prediksi

2.2.13 Python

Python adalah bahasa pemrograman yang ada sebagai pengembangan dari bahasa pemrograman ABC. Python dikembangkan oleh Guido van Rossum pada tahun 1990-an di CWI, Amsterdam. Python adalah bahasa pemrograman dinamis yang mendukung paradigma pemrograman berbasis objek. Salah satu keunggulan Python adalah lisensinya yang bersifat *Open Source* dan tidak bertentangan dengan *General Public License* (GPL), sehingga Python dapat digunakan secara bebas, termasuk untuk tujuan komersial. Selain itu, Python juga menonjol karena memiliki struktur data tingkat tinggi yang efisien [33].



Gambar 2.18 Logo Python [33]

Python adalah bahasa yang sangat serbaguna dan populer dalam komunitas pengembang perangkat lunak karena fleksibilitasnya dan kemampuannya untuk menangani berbagai jenis tugas pemrograman. Kelebihan Python mencakup [33]:

- a. Kemudahan dalam pembelajaran karena memiliki struktur yang sederhana.
- b. Keterbacaan kode yang baik, di mana penggunaan spasi yang 'menjorok' diperlukan untuk menjaga kebersihan kode.
- c. Kode Python umumnya lebih singkat daripada bahasa lain untuk menyelesaikan masalah yang serupa.
- d. Python dapat digunakan di berbagai sistem operasi, termasuk Windows, UNIX, Linux, dan Mac OS.
- e. Python mendukung pemrograman prosedural.
- f. Bahasa ini memiliki banyak *library* dan paket yang dapat digunakan untuk berbagai keperluan khusus.