

BAB II

TINJAUAN PUSTAKA

2.1. Ekspresi Wajah

Secara umum, ekspresi wajah adalah cara komunikasi nonverbal yang kuat untuk mengungkapkan emosi manusia. Ekspresi wajah dapat mencakup berbagai gerakan dan posisi wajah yang mencerminkan perasaan atau emosi yang sedang dialami seseorang (Sugiarno & Ginting, 2019). Ekspresi wajah memiliki sejumlah kegunaan penting dalam konteks sosial dan interpersonal. Ekspresi wajah ini membantu individu untuk mengungkapkan dan berbagi emosi dengan orang lain tanpa kata-kata. Selain itu, ekspresi wajah juga berperan dalam mempengaruhi komunikasi dan interaksi sosial antara individu, seperti memperoleh dukungan emosional, membangun hubungan, atau mengkomunikasikan kebutuhan dan keinginan.

Oleh karena itu berikut adalah penjelasan mengenai ekspresi wajah umum dan makna psikologis yang terkait ekspresi-ekspresi tersebut:

1. Marah

Ekspresi wajah marah ditandai dengan alis yang terangkat, bibir tegang atau tertutup, dan mata yang menyipit. Ekspresi ini mencerminkan rasa kemarahan, frustrasi, atau keinginan untuk melawan atau menghadapi ancaman.

2. Muak

Ekspresi wajah muak biasanya ditandai dengan hidung yang berkerut, bibir yang melengkung ke bawah, dan ekspresi wajah yang menunjukkan ketidaknyamanan atau ketidaksenangan terhadap suatu hal atau situasi.

3. Takut

Ekspresi wajah takut biasanya ditandai dengan alis yang terangkat, mata yang melebar, dan mulut yang terbuka. Ekspresi ini mencerminkan rasa

ketakutan, kecemasan, atau ancaman yang dirasakan oleh individu terhadap suatu hal atau situasi.

4. Senang

Ekspresi wajah senang ditandai dengan mata yang berkerut atau berkedip, senyuman, dan ekspresi wajah yang terlihat bahagia atau puas. Ekspresi ini mencerminkan kegembiraan, kepuasan, atau perasaan positif yang dirasakan oleh individu.

5. Netral

Ekspresi wajah netral ditandai dengan ekspresi yang relatif datar atau tanpa ekspresi emosi yang jelas. Ini adalah ekspresi wajah yang umumnya dianggap tidak menunjukkan emosi tertentu, meskipun individu mungkin tetap memiliki perasaan atau emosi yang sedang dialami.

6. Sedih

Ekspresi wajah sedih biasanya ditandai dengan alis yang tertarik ke bawah, mata yang terlihat lelah atau mengisap dengan air mata, dan bibir yang menggantung atau tertarik ke bawah. Ekspresi ini mencerminkan perasaan sedih, kekecewaan, atau kesedihan yang dirasakan oleh individu.

7. Kaget

Ekspresi wajah kaget biasanya ditandai dengan mata yang melebar, alis yang terangkat, dan mulut yang terbuka. Ekspresi ini mencerminkan perasaan kaget, terkejut, atau tidak siap menghadapi sesuatu yang tak terduga.

2.2. *Artificial Intelligence*

Penelitian ini membangun perangkat lunak berbasis kecerdasan buatan atau *artificial intelligence* (AI). Menurut (Markiewicz & Zheng, 2020) AI adalah sistem yang membuat sebuah komputer memiliki kecerdasan seperti manusia. Dengan

tujuan tertentu AI dapat memberikan sebuah masukan atau hasil yang mirip dengan kecerdasan manusia berdasarkan data yang sudah diberikan.

2.2.1. Machine Learning

Machine learning adalah sub bagian dari AI yang berfokus pada komputer untuk memberikan keputusan dari sebuah masalah tanpa membuat sebuah program yang kompleks. Menurut (Markiewicz & Zheng, 2020) *machine learning* memiliki 3 jenis yakni :

- *Supervised Learning*
Supervised learning dapat dilatih dari sebuah data yang berlabel atau terstruktur. Semakin besar data yang diberikan maka semakin baik *output* yang dikeluarkan. Pada *supervised learning*, sebuah sistem belajar dari label yang diberikan oleh manusia.
- *Unsupervised Learning*
Unsupervised learning dilatih dari sebuah data yang tidak ter label dan terstruktur. Di mana algoritma tersebut akan mencari data dan mencoba membuat data tersebut menjadi terstruktur. Salah satu contoh penerapan *unsupervised learning* adalah *clustering*.
- *Reinforcement Learning*
Reinforcement learning belajar dan memungkinkan melakukan pengoptimalan pada algoritmanya sendiri. Cara belajar algoritma *reinforcement learning* mirip seperti *supervised learning* namun memiliki perbedaan, di mana algoritma tersebut akan mengambil keputusan berdasarkan *reward* yang sudah didapat sebelumnya.

2.2.2. Deep Learning

Deep learning adalah suatu cara untuk mengembangkan komputer untuk belajar menggunakan *neural network* seperti pada jaringan syaraf manusia. Algoritma *deep learning* terdiri dari sekumpulan jaringan yang terhubung dinamakan

neuron. *Deep learning* juga merupakan salah satu algoritma yang paling dekat dengan bagaimana cara otak manusia memproses sebuah informasi melalui jaringan syaraf dan memberikan berbagai macam hasil. *Deep learning* dapat mengembangkan sebuah model dengan berbagai macam data seperti gambar, teks, video, suara. Untuk menghasilkan performa latih yang maksimal *deep learning* memanfaatkan keuntungan dari sebuah GPU. Pada saat ini penggunaan *deep learning* memiliki kemajuan yang sangat pesat seperti pengenalan suara, gambar, mesin penerjemah dan beberapa sistem cerdas yang dapat bekerja lebih baik dari pada manusia (Markiewicz & Zheng, 2020).

2.3. Neural Network

Neural network adalah suatu cara untuk memprogram komputer yang dapat beroperasi mirip dengan otak manusia. Tujuan utama dari *neural network* ini adalah untuk melakukan fungsi yang dapat dilakukan oleh otak manusia seperti pemecahan masalah dan dapat belajar (Suwindra et al., 2021). Menurut (Merchant, 2019) *neural network* memiliki beberapa tipe antara lain:

- *Feedforward Neural Network*
Feedforward Neural Network adalah salah satu tipe *neural network* yang sederhana, di mana data dari *input* akan bergerak satu arah saja. Di mana setelah data melewati *input nodes* dan keluar melalui *output nodes*, dan *feedforward NN* tidak bisa memberikan *feedback* sehingga ketika model mengeluarkan *output* yang salah maka tidak bisa memperbaiki sendiri.
- *Radial Basis Function Neural Network*
Radial Bias Function menghitung jarak dari suatu titik terhadap titik pusat. RBF memiliki dua layer, di mana pada layer pertama fitur akan dikombinasikan dengan *Radial Bias Function* pada *inner layer* dan kemudian keluaran dari fitur akan di pertimbangkan saat menghitung keluaran yang sama pada fungsi berikutnya.
- *Recurrent Neural Network (RNN)*

Prinsip kerja *RNN* dengan menyimpan *output* layer dan memberikannya lagi ke *input* untuk membantu prediksi dari keluaran layer tersebut. *RNN* menggunakan algoritma *backpropagation* untuk memeriksa ulang dan memastikan 99.9% keluarannya benar.

- *Convolutional Neural Network (CNN)*

CNN mirip dengan *feedforward neural network*, di mana sebuah neuron memiliki bobot dan bias yang dapat digunakan untuk belajar. Banyak orang yang menyukai *CNN* untuk *image processing*.

- *Modular Neural Network (MNN)*

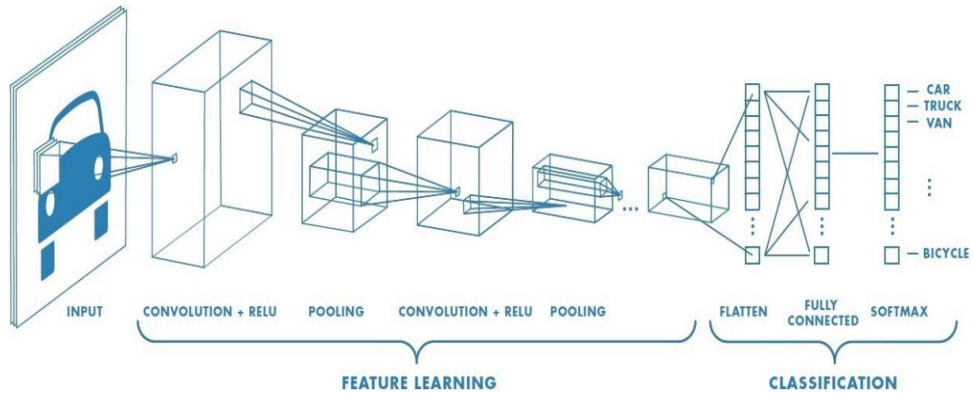
MNN adalah sekumpulan jaringan yang bekerja secara independen sembari berkontribusi terhadap *output*. Setiap jaringan memiliki satu set *input* yang unik dibandingkan dengan jaringan yang lain yang membangun dan menjalankan *sub-tasks*. Keuntungan dari *MNN* adalah mereka memecah proses komputasi ke dalam proses yang kecil di mana dapat melakukan fungsi komputasi yang lebih mudah dan cepat.

2.4. *Convolutional Neural Network*

Fasilitas pengenalan gambar untuk deteksi penyakit yang dibangun saat ini berbasis *Convolutional Neural Network* yang merupakan representasi dari arsitektur jaringan syaraf manusia dan termasuk ke dalam algoritma *deep learning* untuk *computer vision*. *CNN* telah menunjukkan kinerja tinggi pada berbagai tugas yang berhubungan dengan *computer vision*, seperti *image classification*, *image segmentation*, *image retrieval*, *object detection*, *image captioning*, *face recognition*, *pose estimation*, *trafig sign recognition*, *speech processing*, dll. *CNN* terdiri dari *perceptron* yang mengoptimalkan dirinya melalui pembelajaran. Setiap *perceptron* akan menerima *input* dan melakukan operasi (Dewi, 2018).

2.4.1. Arsitektur *Convolutional Neural Network*

Berikut adalah arsitektur *convolutional neural network* yang dapat dilihat pada gambar 2.1



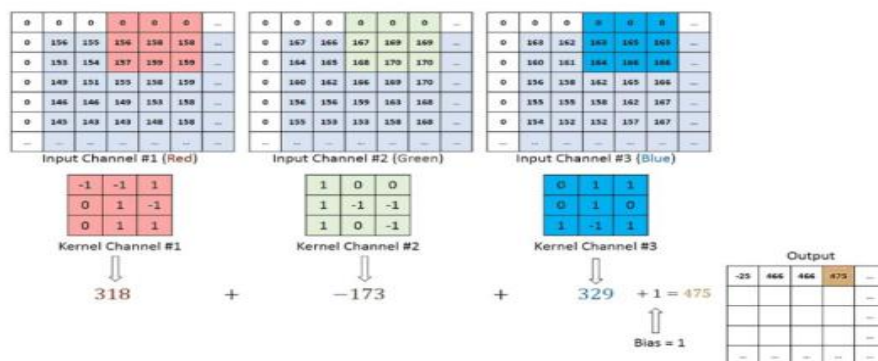
Gambar 2. 1 Arsitektur *Convolutional Neural Network*

Menurut (Bačanin Džakula, 2019) *Convolutional Neural Network* memiliki 3 lapisan yaitu:

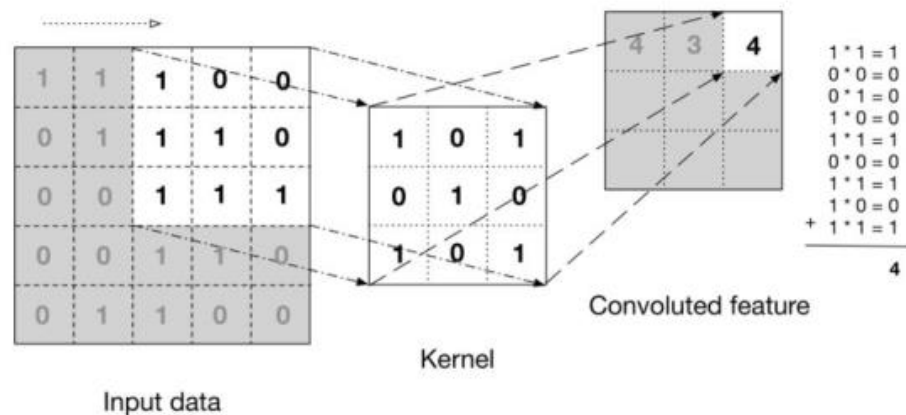
1. *Convolutional Layer*

Convolutional layer memiliki peran yang sangat penting dalam arsitektur CNN. pada layer ini mengaplikasikan sebuah filter (*kernel*) agar dapat belajar. Setiap filter memiliki ukuran yang kecil dan dapat menyebar dan memenuhi volume dari *input*. Biasanya sebuah filter memiliki ukuran 3x3, 5x5, 7x7.

Berikut adalah proses konvolusi pada gambar *grayscale* dan RGB dengan menggunakan ukuran filter 3 x 3.



Gambar 2. 2 Konvolusi Citra RGB



Gambar 2. 4 Konvolusi Citra Grayscale

Pada gambar 2.2 dan 2.3 yang merupakan konvolusi citra (citra *grayscale* dan RGB) dengan menggunakan filter 3x3 yang diterapkan ke dalam gambar *input* untuk mendapatkan *output* atau *feature map*.

Untuk mencari ukuran dari *output* / *feature map* dipengaruhi oleh 3 *hyperparameter* yakni *depth*, *stride* dan *padding*:

- *Depth*
Depth adalah *output* dari volume merepresentasikan ukuran dari filter yang digunakan dalam proses konvolusi. Setiap filter mempelajari sesuatu dari *input* yang berbeda seperti, tepi, warna, dan lainnya.
- *Stride*
Stride adalah jumlah langkah yang dilalui oleh filter pada suatu *input*. Ketika nilai dari *stride* 1 maka filter akan bergerak 1 langkah ketika *stride* bernilai 2 maka filter akan bergerak sejauh 2 langkah.
- *Padding*
Padding memungkinkan kita mengatur ukuran dari *output*. Dengan menerapkan konvolusi pada *input*, dapat mengurangi ukuran *output* yang menyebabkan kehilangan informasi dari citra yang dimiliki, untuk mencegahnya dapat mengisi volume *input* dengan 0 di sekitar

pinggiran citra. Ada 2 pilihan yakni *valid convolution* dan *same convolution*. *Valid convolution* berarti tidak memiliki *padding*, dan *same convolution* berarti memiliki ukuran *output* sama dengan ukuran *input*

Sehingga untuk mencari sebuah *output shape* dapat dilakukan menggunakan persamaan 2.1.

$$n_{out} = \left(\frac{n_{in} + 2 \cdot p - f}{s} \right) + 1$$

Persamaan 2. 1 Rumus Mencari Output Size

Di mana (n_{in}) adalah ukuran *input*, dan (p) adalah jumlah dari *padding*, (f) adalah ukuran dari *filter* dan (s) adalah jumlah dari *stride*.

Setelah ditemukan *output size* atau *output shape* yang akan dilakukan dengan perhitungan dot *product* oleh sebuah filter terhadap bagian *input* yang dilaluinya. Sehingga untuk mencari sebuah *output* atau *feature map* dapat dilakukan dengan persamaan 2.2.

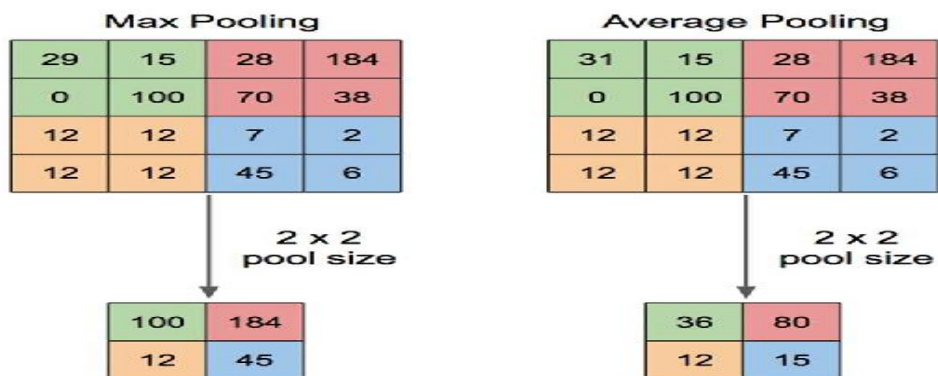
$$a \cdot b = \sum_{i=1}^n a_i \cdot b_i$$

Persamaan 2. 2 Rumus Dot Product

2. *Pooling Layer*

Pooling layer digunakan setelah proses *convolution layer* selesai. Fungsi *pooling layer* adalah untuk mengurangi dimensi pada gambar sehingga dapat mempercepat komputasi karena parameter yang di *update* akan semakin sedikit dan mengatasi *overfitting*.

Penggunaan *pooling* layer umumnya menggunakan filter berukuran 2 x 2. Umumnya *pooling* layer memiliki dua tipe yakni *max pooling* dan *average pooling*. *Max Pooling* bekerja dengan mengambil nilai maksimum dari setiap bagian yang dicakup oleh filter sedangkan *Average Pooling* bekerja dengan mengambil nilai rata-rata dari bagian gambar yang dicakup oleh sebuah filter.



Gambar 2. 5 Proses *Max Pooling* dan *Average Pooling*

Untuk mencari output shape dari pooling layer dapat menggunakan persamaan 2.3.

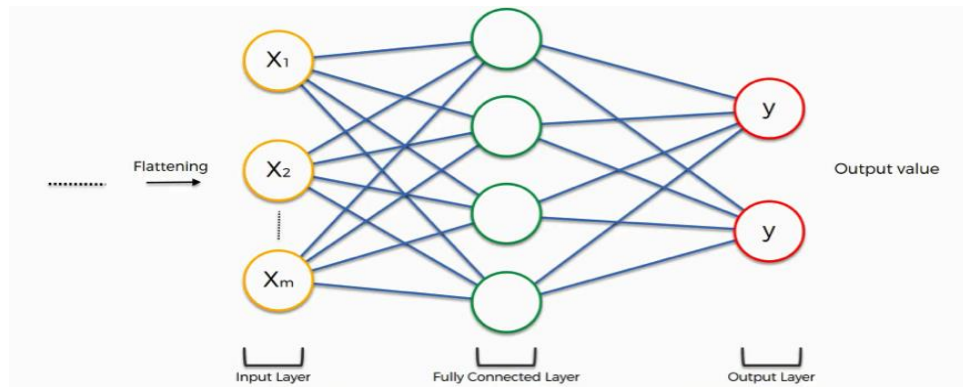
$$\frac{n - p}{s} + 1$$

Persamaan 2. 3 Rumus Mencari Output Size Pada Layer Pooling

Di mana (n) adalah ukuran *input*, dan (p) adalah jumlah dari *padding*, (s) adalah jumlah dari *stride*.

3. *Fully Connected Layer*

Setelah melakukan proses pada layer konvolusi dan *pooling* layer CNN berakhir dengan layer yang terhubung secara penuh atau *fully connected* layer di mana pada bagian ini data akan di ubah ke dalam 1 dimensi atau biasa di sebut juga dengan *flatten*.



Gambar 2. 6 Bentuk Lapisan *Fully Connected Layer*

Gambar diatas dapat dilihat bahwa pada *fully connected*. terjadi inisialisasi bobot, bias, serta penambahan fungsi aktivasi pada setiap perhitungan antar *input* dengan *perceptron*

2.5. Fungsi Aktivasi

Activation Function adalah sebuah fungsi aktivasi yang digunakan pada *neural network* untuk menghitung seluruh dari jumlah bobot dan bias, yang dapat digunakan untuk memutuskan apakah neuron bisa di aktifkan atau tidak(Emanuella, 2022). Fungsi aktivasi yang digunakan pada penelitian kali ini adalah:

- *ReLU (Rectified Linear Unit)*

Rectified linear unit (ReLU) adalah fungsi aktivasi yang banyak digunakan saat ini. Fungsi aktivasi ini biasa digunakan pada *hidden layer* terutama sebagian besar pada *Convolutional Neural Network* (CNN) atau *Deep Learning* (DL)(Feng & Lu, 2019). Rentang nilai dari ReLU $\{0, -\infty\}$. Jika nilai $x < 0$ *output* yang dihasilkan adalah 0 dan jika nilai $x > 0$ *output* yang

dihasilkan adalah nilai x itu sendiri. Sehingga untuk mencari nilai menggunakan fungsi aktivasi *ReLU* dapat dilihat pada persamaan 2.4.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x < 0 \end{cases}$$

Persamaan 2. 4 Fungsi Aktivasi ReLU

- *Softmax*

Softmax function adalah salah satu fungsi aktivasi yang digunakan pada *neural network*. Fungsi ini menghitung distribusi probabilitas dari sebuah *vector*. *Softmax* menghasilkan keluaran dengan rentang 0 dan 1, di mana jumlah probabilitas sama dengan 1. *Softmax* digunakan pada *multiclass classification* dan kebanyakan diimplementasikan pada bagian *output layer*.

2.6. Confusion Matrix

Confusion Matrix adalah sebuah *matrix* yang memvisualisasikan kinerja algoritma klasifikasi menggunakan data *matrix*, dengan membandingkan klasifikasi yang di prediksi dengan klasifikasi aktual dalam bentuk *false positive*, *true positive*, *false negative* dan *true negative*(Desiani, 2022).

Tabel 2. 1 Confusion Matrix

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Pada tabel di atas dapat dijelaskan sebagai berikut:

- TP : Hasil di identifikasi benar dengan nilai *positive*
- FP : Hasil di identifikasi benar dengan nilai *negative*
- FN : Hasil identifikasi salah dengan nilai *positive*
- TN : Hasil identifikasi salah dengan nilai *negative*

Untuk menghitung akurasi dapat menggunakan persamaan 2.6:

$$accuracy = \frac{TruePrediction + FalsePrediction}{TotalData}$$

Persamaan 2. 5 Rumus Mencari Accuracy

2.7. Software yang Digunakan untuk Pengembangan Aplikasi

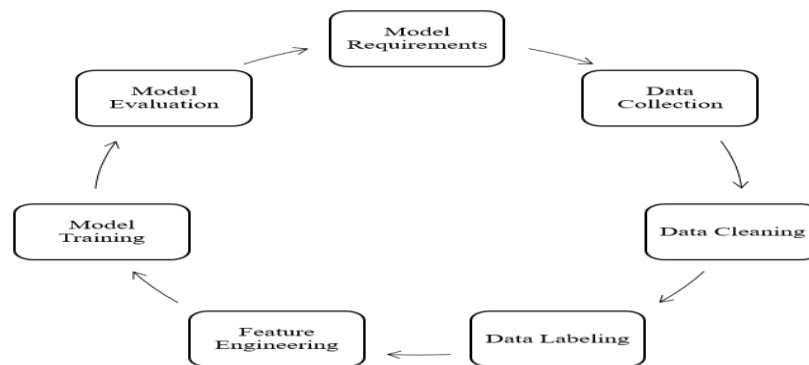
2.7.1. Google Colaboratory

Google Colaboratory biasa di sebut *google colab* adalah salah satu alat yang digunakan untuk mengembangkan proyek *machine learning* baik itu untuk penelitian ataupun edukasi. *Google colab* berbasis *jupyter notebook*, dan dapat di bagi oleh pengguna lain untuk berkolaborasi pada *notebook* yang sama. *Google colab* menyediakan versi *python 2* atau *python 3* yang sudah di konfigurasi dengan pustaka *machine learning* dan *artificial intelligence* seperti *tensorflow*, *matplotlib*, *skitlearn*, keras. *Google collab* berjalan di atas *virtual machine* (VM) yang di integrasikan dengan *google cloud platform*(Nelson & Hoover, 2020).

2.8. Metode Pengembangan Model

2.8.1. *Machine Learning Life Cycle*

Berikut adalah metode pengembangan perangkat lunak menggunakan *machine learning life cycle*



Gambar 2. 7 Machine Learning Life Cycle

Pada tahapan pengembangan perangkat lunak menggunakan *machine learning life cycle* memiliki beberapa tahapan antara lain (Gärtler et al., 2021):

1. *Model Requirements*

Pada fase *model requirements* dilakukan pemilihan model, pengumpulan data, dan jumlah data yang akan diambil kemudian akan diimplementasikan ke dalam *machine learning*. Setelah model yang sudah sesuai kemudian akan diimplementasikan.

2. *Data Collection*

Pada fase *data collection* akan dilakukan pengumpulan data terkait kasus penelitian yang kemudian akan dilakukan pelatihan untuk model.

3. *Data Cleaning*

Data cleaning digunakan untuk memastikan kualitas dari sampel seperti akurasi kelengkapan, konsistensi, keunikan dan integritas. Pada *image processing data cleaning* sangat diperlukan untuk standarisasi ukuran dari suatu gambar.

4. *Data Labeling*

Data labeling adalah proses pelabelan suatu data yang dibantu oleh pakar.

5. *Feature Engineering*

Feature engineering adalah proses untuk mengekstrak, seleksi, mengubah dan memilih fitur dari data mentah ke dalam bentuk yang sesuai untuk model

6. *Model Training*

Model training mencakup aktivitas untuk melatih dan memilih model *machine learning* kemudian menyesuaikan *hyperparameter* pada data yang sudah dikumpulkan, dibersihkan dan dilabelkan.

7. *Model Evaluation*

Model evaluation pada tahapan ini akan mengkonfirmasi bahwa suatu model dapat memenuhi persyaratan dalam sebuah kasus. Pada kasus kali ini akan menggunakan teknik *supervised learning* di mana model akan di evaluasi berdasarkan kesalahan yang model perbuat dalam melakukan prediksi.

2.9. Penelitian Terkait

Ada beberapa penelitian terkait dengan penggunaan teknik convolutional neural network yang dapat dilihat pada tabel 2.2.

Tabel 2. 2 Penelitian Terdahulu

No	Penulis	Judul	Ringkasan
1	(Amaanullah et al., 2022)	Implementasi convolutional Neural Network Untuk Deteksi Emosi Melalui Wajah	Hasil dari pengujian didapat 508 citra yang berhasil dideteksi menggunakan metode CNN (Convolutional Neural Network). Pada 508 citra tersebut sudah diklasifikasikan berdasarkan kategori yang sudah ditentukan. Dapat disimpulkan bahwa metode CNN bisa digunakan untuk mendeteksi wajah manusia dengan akurasi sebesar 81.92% untuk pelatihan dan 81.69% untuk pengujian, akan tetapi masih memiliki tingkat error yang cukup besar yaitu 75,60% berdasarkan hasil validation loss. Ekspresi yang diberikan harus jelas dengan penerangan yang tepat sehingga sistem mampu

No	Penulis	Judul	Ringkasan
			mendapatkan hasil yang lebih baik
2	(Azizi, 2021)	Deteksi Emosi Menggunakan Citra Ekspresi Wajah Secara Otomatis	Model terbentuk dari hasil training dengan menggunakan batch size 64 dengan pengujian 3 optimasi yang berbeda dan dengan epoch yang berbeda. Perbandingan dataset yang digunakan yaitu 80% training, 10% validasi, dan 10% test dengan nilai akurasi sebesar 82%.
3	(Azhari, 2021)	Implementasi Algoritma convolutional Neural Network Dalam Deteksi Emosi Manusia Berdasarkan Ekspresi Wajah.	Kesimpulan 1. Deep Learning dapat mendeteksi dan mengklasifikasi objek dari input yang berupa device webcam dengan cukup baik serta memiliki tingkat keakurasian yang terbilang cukup bagus antara rentang 67-83%.
4	(Darmawan & Supeno, 2022)	Klasifikasi Ekspresi Wajah Menggunakan Convolutional Neural Network (studi kasus kuliah daring)	Dalam penelitian menghasilkan tingkat akurasi dan testing dalam melakukan klasifikasi ekspresi wajah untuk mengetahui antusiasme siswa/mahasiswa dalam

No	Penulis	Judul	Ringkasan
			<p>pembelajaran kuliah daring sebesar 1% lebih besar tidak antusias dibandingkan dengan yang antusias. Dari percobaan yang telah dilakukan didapatkan hasil precission sebesar 68% recall sebesar 68%, dan akurasi sebesar 64% dan dapat dinilai bekerja dengan baik untuk klasifikasi ekspresi wajah dengan tingkat poor classification.</p>
5	(Karnila et al., 2019)	Face Recognition using Content Based Image Retrieval for Intelligent Security	<p>Intelligent face recognition application using content based image retrieval and Real Time Face Recognition. Application design was carried out quite good since the accuracy of detection fairly good. When face has been searching is not in the database the system that face data is not found. The accuracy of face detection is quite good which from 25 test produce the average of more than 75%</p>

No	Penulis	Judul	Ringkasan
6	(Setyawan & Arkhiansyah, 2011)	Pencarian Citra Berbasis Pengenalan Wajah untuk Absensi mengajar Dosen IBI Darmajaya	Dari hasil penelitian yang diperoleh, dapat disimpulkan bahwa sistem dapat melakukan proses face detection sebesar 85.5 % dan proses face recognition sebesar 60 % secara real time dan hasil face recognition dari masing-masing face belum digunakan sebagai key database dosen mengajar