

SKRIPSI

KLASIFIKASI EKSPRESI WAJAH PESERTA DIDIK DALAM PROSES PEMBELAJARAN DI KELAS

Disusun dan diajukan oleh:

**AMIQATUN NASYATI YUSRI
D121 20 1080**



**PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2024**

PENGESAHAN SKRIPSI

KLASIFIKASI EKSPRESI WAJAH PESERTA DIDIK DALAM PROSES PEMBELAJARAN DI KELAS

Disusun dan diajukan oleh

Amiqatun Nasyati Yusri
D121 20 1080

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka
Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika
Fakultas Teknik Universitas Hasanuddin
Pada tanggal 19 Juli 2024
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing,



Dr. Amil Ahmad Ilham, S.T., M.IT.
NIP 19731010 199802 1 001



Ketua Program Studi,

Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM., ASEAN.Eng.
NIP 19750716 200212 1 004

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Amiqatun Nasyati Yusri
NIM : D121201080
Program Studi : Teknik Informatika
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Klasifikasi Ekspresi Wajah Peserta Didik dalam Proses Pembelajaran di Kelas

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitnya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasikan oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 19 Juli 2024

Yang Menyatakan



Amiqatun Nasyati Yusri

ABSTRAK

AMIQATUN NASYATI YUSRI. *Klasifikasi Ekspresi Wajah Peserta Didik Dalam Proses Pembelajaran di Kelas* (dibimbing oleh Amil Ahmad Ilham)

Ekspresi wajah merupakan salah satu bentuk komunikasi non-verbal yang penting dalam mengungkapkan emosi manusia, seperti, senang, sedih, terkejut, marah, takut, dan jijik. Dalam konteks pendidikan, pemahaman ekspresi wajah peserta didik memiliki peran krusial dalam evaluasi pembelajaran, karena dapat memberikan informasi mengenai fokus dan pemahaman peserta didik terhadap materi yang diajarkan.

Penelitian ini bertujuan untuk mengembangkan model klasifikasi ekspresi wajah peserta didik dalam proses pembelajaran di kelas, yang terbagi kedalam enam kelas ekspresi wajah. Selain itu, penelitian ini juga bertujuan untuk mengukur performa model dengan melakukan beberapa percobaan dengan membandingkan nilai *hyperparameter batch size, epoch, learning rate*, dan penggunaan ‘*class_weight*’.

Sistem ini dikembangkan menggunakan metode CNN yang memadukan dua model melalui *functional API Keras*. Model pertama adalah model citra wajah dengan input berukuran *96x96 pixel* dan *1 channel grayscale*, sedangkan model kedua adalah model fitur geometri wajah. Fitur-fitur geometri tersebut dihasilkan dari proses perhitungan fitur wajah seperti alis, mata, dan mulut, dimana setiap *landmark* wajah diperoleh menggunakan *pretrained dlib*. Proses perhitungan tersebut menghasilkan 9 fitur yang kemudian direduksi menjadi 2 dimensi menggunakan PCA. Total data yang digunakan sebanyak 2.069 citra wajah yang terbagi ke dalam enam kelas ekspresi, bahagia, sedih, netral, terkejut, marah, dan lelah.

Hasil penelitian menunjukkan bahwa model yang dikembangkan mampu mengklasifikasikan ekspresi wajah peserta didik dengan akurasi 92% pada data validasi dan 95% pada data *testing*. *Hyperparameter* yang digunakan meliputi *batch size* 12, *epoch* 50, dan *learning rate* 0,001, tanpa menggunakan ‘*class_weight*’.

Kata Kunci: Ekspresi Wajah, Geometri Wajah, Klasifikasi, *Convolutional Neural Network from scratch*, PCA

ABSTRACT

AMIQATUN NASYATI YUSRI. *Classification of Students' Facial Expressions During the Learning Process in the Classroom* (supervised Amil Ahmad Ilham)

Facial expressions are a crucial form of non-verbal communication that convey human emotions such as happiness, sadness, surprise, anger, fear, and disgust. In an educational context, understanding students' facial expressions plays a vital role in evaluating the learning process, as it can provide information regarding students' focus and comprehension of the material being taught.

This research aims to develop a classification model for students' facial expressions during classroom learning, categorized into six facial expression classes. Additionally, the study measures the model's performance through several experiments, comparing hyperparameter values such as batch size, epochs, learning rate, and the use of 'class_weight'.

This system was developed using the CNN method, integrating two models through the Keras functional API. The first model is a facial image model with an input size of 96x96 pixels and 1 grayscale channel, while the second model is a facial geometric features model. These geometric features are derived from the calculation of facial features such as eyebrows, eyes, and mouth, where each facial landmark is obtained using pretrained dlib. This calculation process produces 9 features, which are then reduced to 2 dimensions using PCA. A total of 2,069 facial images were used, divided into six expression classes: happy, sad, neutral, surprised, angry, and tired.

The research findings demonstrate that the developed model achieves 92% accuracy on validation data and 95% accuracy on testing data. Hyperparameters used include a batch size of 12, 50 epochs, and a learning rate of 0.001, without employing 'class_weight'.

Keywords: Facial Expression, Facial Geometry, Classification, Convolutional Neural Network from scratch, PCA

DAFTAR ISI

PENGESAHAN SKRIPSI	i
PERNYATAAN KEASLIAN	ii
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI	v
DAFTAR GAMBAR	vi
DAFTAR TABEL	viii
DAFTAR SINGKATAN DAN ARTI SIMBOL	ix
DAFTAR LAMPIRAN	x
KATA PENGANTAR	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Ruang Lingkup	3
BAB II TINJAUAN PUSTAKA	4
2.1 Ekspresi Wajah	4
2.2 Citra Digital	7
2.3 <i>Image Resizing</i>	13
2.4 <i>Histogram Equalization</i>	13
2.5 Dlib	15
2.6 <i>Geometrical Feature Extraction</i>	16
2.7 <i>Principal Component Analysis (PCA)</i>	19
2.8 TensorFlow	22
2.9 <i>Deep Learning</i>	23
2.10 CNN	24
2.11 <i>Confusion Matrix</i>	29
BAB III METODE PENELITIAN	32
3.1 Waktu dan Lokasi Penelitian	32
3.2 Benda Uji dan Alat	32
3.3 Tahapan Penelitian	33
3.4 Teknik Pengambilan Data	34
3.5 Perancangan dan Implementasi Sistem	35
BAB IV HASIL DAN PEMBAHASAN	55
4.1. Hasil Penelitian	55
4.2. Pembahasan	73
BAB V PENUTUP	81
5.1 Kesimpulan	81
5.2 Saran	81
DAFTAR PUSTAKA	82
LAMPIRAN	85

DAFTAR GAMBAR

Gambar 1. Contoh ekspresi bahagia.....	4
Gambar 2. Contoh ekspresi marah.....	5
Gambar 3. Contoh ekspresi terkejut.....	5
Gambar 4. Contoh ekspresi sedih.....	6
Gambar 5. Contoh ekspresi lelah	6
Gambar 6. Contoh ekspresi netral.....	7
Gambar 7. Contoh citra biner.....	8
Gambar 8. Contoh citra grayscale dengan kedalaman 8 bit (256 kombinasi warna keabuan)	9
Gambar 9. Format 8 bit	10
Gambar 10. 8 bit truecolor	11
Gambar 11. Contoh citra warna 8 bit dengan palet.....	11
Gambar 12. Format 16 bit	11
Gambar 13. 16 bit highcolor	12
Gambar 14. Contoh citra warna 16 bit	12
Gambar 15. Contoh citra warna 24 bit	13
Gambar 16. Contoh implementasi histogram equalization	14
Gambar 17. Visualisasi 68 koordinat facial landmark dari dataset iBUG 300-W	16
Gambar 18. Facial landmarks dengan 14 titik pada fitur wajah	17
Gambar 19. Ekstraksi fitur geometri wajah	18
Gambar 20. Kerangka konseptual yang menggambarkan efektivitas penggunaan PCA untuk pemilihan fitur	19
Gambar 21. Deep learning sebagai cabang ilmu artificial intelligence dan machine learning	23
Gambar 22. Arsitektur CNN	24
Gambar 23. Contoh konvolusi citra ukuran 5x5x1 dengan 3x3x1 kernel ...	25
Gambar 24. Contoh Max pooling layer 3x3 pada matriks 5x5	26
Gambar 25. Contoh fully connected layer	29
Gambar 26. Lokasi penelitian	32
Gambar 27. Tahapan Penelitian	33
Gambar 28. Ilustrasi pengambilan video	35
Gambar 29. Rancangan Sistem	36
Gambar 30. Contoh deteksi wajah pada data video	37
Gambar 31. Contoh hasil crop wajah.....	38
Gambar 32. Konversi citra RGB ke grayscale	39
Gambar 33. Ilustrasi dan grafik citra wajah setelah melalui histogram equalization	40
Gambar 34. Contoh hasil konversi matriks 3x3 dari citra grayscale ke histogram equalization	41
Gambar 35. Ilustrasi deteksi landmark wajah	42
Gambar 36. 14 titik landmark yang digunakan untuk fitur geometri wajah	42
Gambar 37. Model arsitektur CNN.....	48
Gambar 38. Contoh proses konvolusi pada citra 6x6	51
Gambar 39. Proses pergeseran filter	51

Gambar 40. Proses fungsi aktivasi ReLU	52
Gambar 41. Proses <i>batch normalization</i>	52
Gambar 42. Proses max pooling	52
Gambar 43. Proses mengubah data menjadi vektor satu dimensi.....	53
Gambar 44. Arsitektur CNN	56
Gambar 45. Hasil training dengan batch size 8.....	59
Gambar 46. Hasil training dengan batch size 12.....	60
Gambar 47. Hasil training dengan batch size 14.....	60
Gambar 48. Hasil training dengan batch size 16.....	61
Gambar 49. Hasil training dengan batch size 32.....	62
Gambar 50. Hasil training dengan batch size 64.....	63
Gambar 51. Hasil training dengan epoch 50	64
Gambar 52. Hasil training dengan epoch 60	64
Gambar 53. Hasil training dengan epoch 70	65
Gambar 54. Hasil training dengan learning rate 0,01	66
Gambar 55. Hasil training dengan learning rate 0,001	67
Gambar 56. Hasil training dengan learning rate 0,0001	68
Gambar 57. Hasil training dengan menggunakan 'class_weight'	69
Gambar 58. Hasil training tanpa menggunakan 'class_weight'	69
Gambar 59. Hasil evaluasi menggunakan confusion matrix.....	71

DAFTAR TABEL

Tabel 1. Confusion matrix dari suatu klasifikasi.....	30
Tabel 2. Nilai skala keabuan, frekuensi, dan distribusi kumulatif	40
Tabel 3. Nilai keabuan hasil	41
Tabel 4. Contoh koordinat hasil ekstraksi landmark wajah	42
Tabel 5. Contoh fitur geometri wajah	45
Tabel 6. Rata-rata nilai fitur geometri pada setiap label	55
Tabel 7. Hasil pengujian hyperparameter batch size.....	74
Tabel 8. Hasil pengujian hyperparameter epoch	75
Tabel 9. Hasil pengujian hyperparameter learning rate	77
Tabel 10. Hasil pengujian hyperparameter penggunaan class_weight.....	79

DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
CNN	<i>Convolutional Neural Network</i>
3D	<i>Three-dimensional</i>
2D	<i>Two-dimensional</i>
B&W	<i>Black and White</i>
RGB	<i>Red, Green, and Blue</i>
HE	<i>Histogram Equalization</i>
AHE	<i>Adaptive Histogram Equalization</i>
CLAHE	<i>Contrast Limited Adaptive Histogram Equalization</i>
PCA	<i>Principal Component Analysis</i>
EVD	<i>Value Decomposition</i>
CUDA	<i>Compute Unified Device Architecture</i>
σ	<i>Varian</i>
API	<i>Application Programming Interface</i>
MLP	<i>Multilayer Perceptron</i>
ReLU	<i>Rectified Linear Unit</i>

DAFTAR LAMPIRAN

Lampiran 1. Contoh Dataset	86
Lampiran 2. Lampiran Source Code Program	89

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah SWT karena atas berkat dan rahmat-Nya sehingga dapat menyelesaikan tugas akhir dengan judul “Klasifikasi Ekspresi Wajah Peserta Didik dalam Proses Pembelajaran di Kelas” sebagai salah satu persyaratan akademik untuk menyelesaikan program Strata-1 di Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari banyak kesulitan dan kendala yang dihadapi saat penyusunan tugas akhir ini. Dalam prosesnya, penulis memperoleh banyak bantuan, dukungan, dan bimbingan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan terima kasih kepada:

1. Allah SWT. atas berkat, rahmat, dan petunjuk-Nya sehingga penulis dapat menyelesaikan tugas akhir ini.
2. Orang tua dan saudara penulis yang senantiasa mendoakan, memberikan semangat, dan bantuan materi maupun non-materi kepada penulis.
3. Keluarga besar penulis yang selalu mendukung, menyemangati, dan mendoakan penulis.
4. Bapak Dr. Amil Ahmad Ilham, S.T., M.IT., selaku pembimbing, yang senantiasa menyediakan waktu, tenaga, pikiran, dan perhatian yang luar biasa dalam mengarahkan penulis untuk menyelesaikan tugas akhir.
5. Segenap Dosen dan Staff Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah banyak membantu penulis selama masa perkuliahan.
6. Ibu Andi Hutami Endang, yang senantiasa membimbing, membantu, menyemangati, dan menghibur penulis. Kata-kata terima kasih tak akan pernah cukup untuk mengungkapkan rasa terima kasih penulis kepada ibu Endang.
7. Keluarga Lab CCIS yang senantiasa membantu, memberikan motivasi, saran, dan membimbing penulis, terkhusus Rifqy, Alif, dan Dimas yang senantiasa mendengar curhatan penulis.
8. Teman-teman Ngok yang telah menjadi teman yang baik, menyenangkan, memberikan warna di dunia perkuliahan penulis, dan juga selalu meladeni lelucon penulis.
9. Kak Yuli, Kak Dipa, dan Bos Iyang, yang senantiasa membantu penulis.
10. Nawang, Difa, Alpi, Sutri, Inung, dan Ica, yang selalu menghibur, mendukung, menyemangati, dan menjadi keluarga penulis.
11. Teman-teman Phoenix yang selalu menjadi rumah kedua bagi penulis.
12. Teman-teman KKN Posko Dannuang yang telah memberi pengalaman berkesan kepada penulis selama KKN
13. Para teman-teman Rezolver 20 yang telah kebersamai penulis selama masa perkuliahan.
14. Serta berbagai pihak atas segala dukungan dan bantuannya yang tidak dapat penulis tuliskan satu persatu.

Penulis berharap semoga Allah SWT. berkenan membalas segala bantuan dan dukungan dari semua pihak yang membantu penulis dalam menyelesaikan tugas akhir ini. Penulis menyadari bahwa tugas akhir ini masih banyak kekurangan dan jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan segala bentuk saran serta masukan yang membangun dari berbagai pihak. Semoga tugas akhir ini dapat berguna bagi para pembaca.

Gowa, XX Juli 2024

Amiqatun Nasyati Yusri

BAB I

PENDAHULUAN

1.1 Latar Belakang

Ekspresi wajah merupakan salah satu metode paling efektif bagi manusia dalam mengungkapkan emosi (Muttaqiin dkk, 2023). William Shakespeare, seorang sastrawan Inggris, mengungkapkan bahwa ekspresi wajah adalah sebuah “buku” yang memungkinkan membaca perasaan dan pikiran seseorang (Amda dan Fitriyani, 2018). Ekspresi wajah merupakan bentuk komunikasi non-verbal yang dihasilkan dari satu atau lebih gerakan atau posisi otot pada wajah yang dapat menyampaikan keadaan emosi seseorang seperti, senang, sedih, terkejut, marah, takut, dan jijik (Seandrio dkk, 2021). Bentuk dari ekspresi tersebut merupakan hasil gerakan otot dari bagian wajah seperti alis, mata, dan mulut (Cahyo dkk, 2023).

Dalam kegiatan belajar mengajar, pemahaman ekspresi wajah peserta didik memiliki peran penting dalam evaluasi pembelajaran, karena memberikan informasi mengenai kefokusannya peserta didik dalam memahami materi pembelajaran. Kualitas dan kuantitas pembelajaran peserta didik sangat dipengaruhi oleh suasana emosional yang dialami selama proses belajar mengajar. Emosi positif dapat mempercepat pemahaman materi dan menciptakan hasil belajar yang optimal, sementara emosi negatif dapat menghambat pemahaman materi dan mengurangi efektivitas proses pembelajaran (Seandrio dkk, 2021).

Dalam pembelajaran di kelas, guru berkonsentrasi mengajar sehingga tidak memperhatikan ekspresi peserta didik yang dapat mengindikasikan ketidakfokusan pada pelajaran (Irawan dkk, 2019). Selain itu, jumlah peserta didik yang banyak membuat guru kesulitan untuk memperhatikan secara individual. Dengan menerapkan teknologi pengenalan ekspresi wajah dalam konteks pembelajaran di kelas, dapat membantu dalam mengevaluasi metode pembelajaran, meningkatkan interaksi guru terhadap peserta didik, dan meningkatkan kualitas pendidikan.

Teknologi pengenalan ekspresi wajah dapat dilakukan dengan menggunakan metode deep learning, terutama metode Convolutional Neural Network (CNN). Dalam metode ini, fitur wajah dapat dideteksi melalui serangkaian proses konvolusi yang diterapkan pada citra (Alamsyah & Pratama, 2020). Metode CNN telah menunjukkan hasil yang signifikan dalam pengenalan citra digital dengan kemampuannya mengklasifikasi objek dan menghasilkan akurasi yang tinggi. CNN memiliki beragam pengaplikasian seperti klasifikasi gambar, deteksi objek, pelacakan objek, estimasi pose, deteksi teks dan pengenalan, pelabelan adegan dan bahasa alami pemrosesan. Terdapat beberapa arsitektur yang sering digunakan dalam CNN. Seperti LeNet 5, AlexNet, VGGNet, GoogLeNet, dan ResNet. Setiap arsitektur memiliki kelebihan dan kelemahan yang masing-masing. Pemilihan arsitektur dalam CNN sangat berpengaruh terhadap hasil klasifikasi suatu objek (Saputri dkk, 2022).

Beberapa penelitian lain telah menunjukkan bahwa Convolutional Neural Network (CNN) telah terbukti memiliki performa yang baik dalam pengenalan objek dan ekspresi wajah, karena kemampuan CNN dalam mempelajari dataset yang besar dan menerapkan metode pelatihan jaringan yang kompleks (Muttaqiin dkk, 2023).

Berdasarkan penjelasan yang telah diuraikan, penulis mengusulkan judul penelitian “Klasifikasi Ekspresi Wajah Peserta Didik dalam Proses Pembelajaran di Kelas”, menggunakan metode CNN, sebagai langkah untuk memanfaatkan teknologi pengenalan ekspresi wajah dalam konteks pembelajaran di kelas dengan fokus pada fitur wajah seperti alis, mata, dan mulut. Penelitian ini diharapkan dapat memberikan kontribusi dalam meningkatkan pemahaman tentang interaksi peserta didik dengan pengajar dalam lingkungan pembelajaran yang lebih baik.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan, maka rumusan masalah dalam penelitian ini yaitu:

1. Bagaimana mengimplementasikan metode CNN dalam melakukan klasifikasi ekspresi wajah peserta didik dalam mengikuti pelajaran?
2. Bagaimana kinerja sistem dalam melakukan klasifikasi ekspresi wajah peserta didik dalam mengikuti pelajaran?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Untuk mengimplementasikan metode CNN dalam melakukan klasifikasi ekspresi wajah peserta didik dalam mengikuti pelajaran.
2. Untuk mengetahui kinerja sistem dalam melakukan klasifikasi ekspresi wajah peserta didik dalam mengikuti pelajaran.

1.4 Manfaat Penelitian

Melalui penelitian ini, diharapkan dapat memberikan manfaat antara lain:

1. Penelitian ini diharapkan dapat membantu mengidentifikasi emosi dan respons emosional peserta didik terhadap materi pembelajaran, metode pengajaran, dan lingkungan kelas.
2. Penelitian ini diharapkan dapat memberikan kontribusi pada bidang psikologi pendidikan dengan memahami bagaimana ekspresi wajah dapat mencerminkan proses mental dan emosional selama pembelajaran.

1.5 Ruang Lingkup

Ruang lingkup dalam penelitian ini adalah sebagai berikut:

1. Data yang digunakan adalah citra wajah tampak depan dan tidak memakai kacamata, dan akan terbatas pada kelas 1 hingga kelas 3 di SD Islam Athirah Makassar.

BAB II TINJAUAN PUSTAKA

2.1 Ekspresi Wajah

Wajah manusia berperan sebagai pusat ekspresi, pengenalan, dan komunikasi (Guntoro dkk, 2022). Ekspresi adalah upaya seseorang untuk menyampaikan perasaannya (emosi) sebagai reaksi terhadap situasi tertentu, baik dari dalam maupun luar dirinya, yang ditunjukkan melalui perubahan biologis, fisiologis, dan berbagai kecenderungan tindakan (sikap dan perilaku) yang berorientasi pada tujuan (Setiawan dkk, 2022). Secara umum, psikolog membagi ekspresi wajah menjadi enam kategori dasar, yaitu marah, jijik, takut, bahagia, sedih, dan terkejut (Ghimire *and* Lee, 2013).

Mengenali emosi seseorang dapat membantu menentukan aktivitas yang dapat meningkatkan suasana hati, terutama dalam kegiatan belajar mengajar. Stabilitas emosi sangat penting untuk keberhasilan proses belajar mengajar, karena emosi mempengaruhi kemampuan siswa untuk fokus dan berpikir dengan baik (Setiawan dkk, 2021).

2.1.1 Ekspresi bahagia

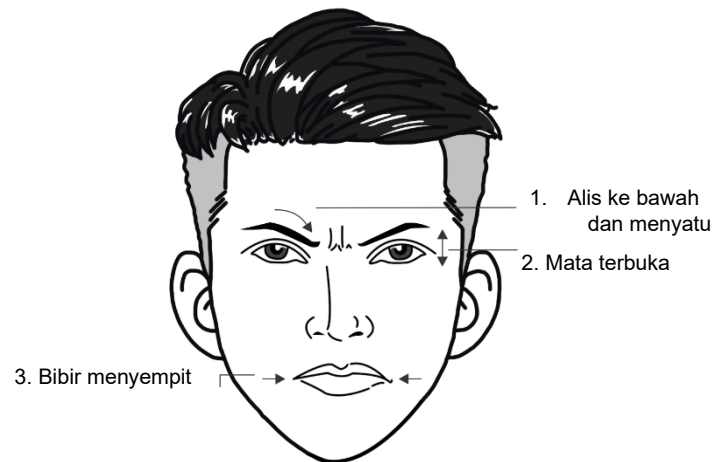
Ekspresi bahagia ditunjukkan melalui otot pipi dan tepi atau sudut bibir bergerak naik (Amda dan Fitriyani, 2016) (Antonius, 2019).



Gambar 1. Contoh ekspresi bahagia

2.1.2 Ekspresi marah

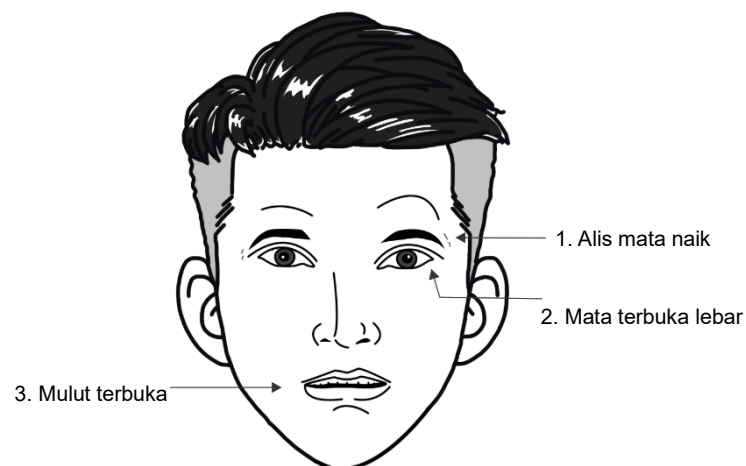
Ekspresi marah ditunjukkan melalui sisi alis bagian dalam yang menyatu dan condong ke bawah, bibir yang menyempit, dan pandangan mata yang menajam (Amda & Fitriyani, 2016) (Antonius, 2019).



Gambar 2. Contoh ekspresi marah

2.1.3 Ekspresi terkejut

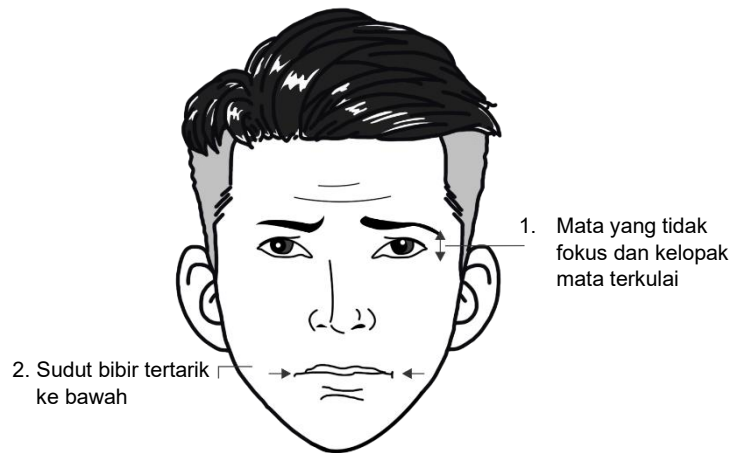
Ekspresi terkejut ditunjukkan dari alis mata yang naik, mata terbuka lebar dan mulut terbuka secara refleks (Amda & Fitriyani, 2016) (Antonius, 2019).



Gambar 3. Contoh ekspresi terkejut

2.1.4 Ekspresi sedih

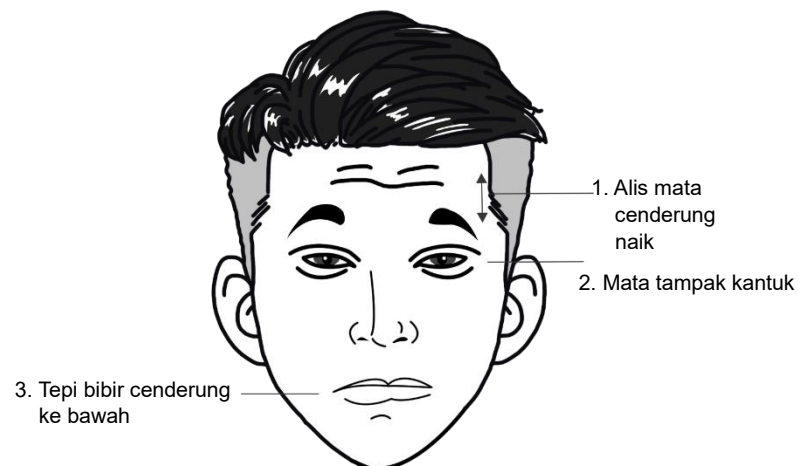
Ekspresi sedih ditunjukkan dari mata yang tidak fokus, sudut bibir tertarik ke bawah, dan kelopak mata terkulai (Amda & Fitriyani, 2016).



Gambar 4. Contoh ekspresi sedih

2.1.5 Ekspresi lelah

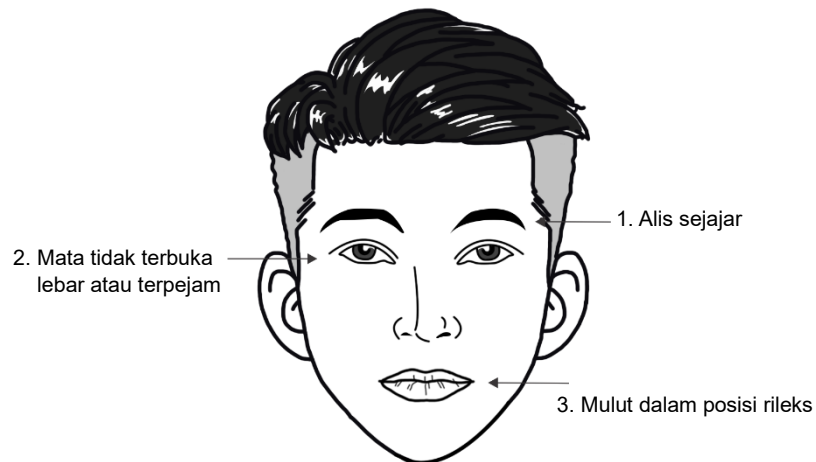
Ekspresi lelah ditunjukkan dengan alis mata cenderung naik, mata tampak kantuk atau berat, dan tepi bibir cenderung ke bawah.



Gambar 5. Contoh ekspresi lelah

2.1.6 Ekspresi netral

Ekspresi netral ditunjukkan dengan alis yang sejajar, mata yang tidak terlalu terbuka lebar atau terpejam, dan mulut berada dalam posisi relatif rileks tanpa senyuman atau tampak tertekan.



Gambar 6. Contoh ekspresi netral

2.2 Citra Digital

Citra digital didefinisikan sebagai fungsi dua dimensi $f(x, y)$ yang memiliki M baris dan N kolom, di mana x dan y adalah koordinat spasial, dan amplitudo f pada titik koordinat (x, y) disebut intensitas atau tingkat keabuan citra pada titik tersebut. Citra digital dihasilkan melalui proses sampling dan kuantisasi, yang menentukan ukuran *pixel* dan tingkat kecerahan. Citra digital terdiri dari elemen-elemen yang disebut *pixel*, dengan setiap *pixel* memiliki lokasi dan nilai tertentu. Nilai *pixel* pada citra digital memiliki rentang tertentu, biasanya dari 0 hingga 255, tergantung pada jenis warnanya. Citra dengan nilai *pixel* ini dikategorikan sebagai citra integer (Gonzales and Wood, 2018).

Citra digital dapat direpresentasikan dalam bentuk matriks dua dimensi, di mana setiap elemen matriks mewakili satu *pixel* dalam citra. Matriks ini memiliki M baris dan N kolom sesuai dengan dimensi citra (Gonzales and Wood, 2018).

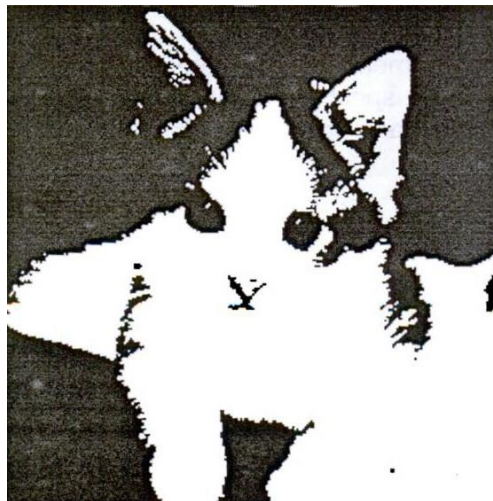
$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1, N-1) \end{bmatrix} \quad (1)$$

Di dalam matriks ini, $f(i, j)$ adalah nilai intensitas atau tingkat keabuan dari *pixel* pada koordinat (i, j) . Nilai ini biasanya berada dalam rentang 0 hingga 255 untuk gambar 8 bit.

Berikut ini adalah jenis-jenis citra berdasarkan nilai *pixel*-nya:

2.2.1 Citra biner

Citra biner adalah citra digital yang hanya memiliki dua nilai *pixel*, yaitu hitam dan putih. Citra biner juga dikenal sebagai citra B&W (*black and white*) atau citra monokrom. Untuk mewakili setiap *pixel* dalam citra biner, hanya diperlukan 1 bit, dengan nilai 0 biasanya mewakili hitam dan nilai 1 mewakili putih. Citra biner sering kali dihasilkan dari proses pengolahan seperti segmentasi, *thresholding*, morfologi, atau *dithering* (Putra, 2010).



Gambar 7. Contoh citra biner
Sumber: (Putra, 2010)

2.2.2 Citra *grayscale*

Citra *grayscale* adalah citra digital yang hanya memiliki satu nilai kanal pada setiap *pixel*-nya, dengan nilai *Red*, *Green*, dan *Blue* yang sama. Nilai ini menunjukkan tingkat intensitas dari hitam, keabuan, hingga putih. Tingkatan keabuan ini mencakup spektrum warna dari hitam hingga hampir putih (Putra, 2010). Jumlah warna tergantung pada jumlah bit yang disediakan di memori. Citra dengan 2 bit dapat mewakili 4 warna, citra dengan 3 bit dapat mewakili 8 warna, dan seterusnya. Semakin besar jumlah bit yang disediakan di memori, semakin halus gradasi warna yang terbentuk (Florestiyanto dkk, 2022).



Gambar 8. Contoh citra *grayscale* dengan kedalaman 8 bit
(256 kombinasi warna keabuan)
Sumber: (Putra, 2010)

2.2.3 Citra warna

Citra warna adalah jenis citra digital yang menyimpan informasi tentang warna setiap *pixel* dalam gambar. Setiap *pixel* diwakili oleh kombinasi dari tiga warna dasar yaitu *Red*, *Green*, dan *Blue*. Model warna ini dikenal sebagai model RGB. Nilai intensitas dari setiap warna dasar menentukan warna akhir yang ditampilkan pada setiap *pixel*. Dengan menggabungkan berbagai tingkat intensitas dari ketiga warna dasar, citra mampu merepresentasikan berbagai spektrum warna yang dapat dilihat oleh mata manusia.

Dalam proses konversi citra warna (RGB) ke citra keabuan (*grayscale*) masing-masing diberi bobot perspektif yang didasarkan pada sensitivitas mata manusia terhadap berbagai warna, di mana mata manusia lebih peka terhadap warna hijau dibandingkan merah, dan lebih peka terhadap merah dibandingkan biru. Berikut adalah formula konversi dari RGB ke *grayscale* (Florestiyanto dkk, 2022).

$$Grayscale = 0,299R + 0,587G + 0,114B \quad (2)$$

dimana,

R : Komponen merah

G : Komponen hijau

B : Komponen biru

Citra warna dapat dibedakan berdasarkan kedalaman bit yang digunakan untuk menyimpan informasi warna. Kedalaman bit menentukan jumlah warna yang dapat direpresentasikan dalam sebuah citra serta kualitas dan akurasi warna yang ditampilkan. Tiga kategori umum berdasarkan kedalaman bit adalah sebagai berikut.

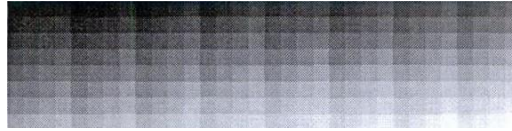
2.2.3.1 Citra warna (8 bit)

Citra warna (8 bit) memiliki setiap *pixel* yang diwakili oleh 8 bit, dengan jumlah warna maksimum yang dapat digunakan adalah 256 warna. Ada dua jenis citra warna 8 bit. Pertama, citra warna 8 bit yang menggunakan palet warna 256, di mana setiap paletnya memiliki pemetaan nilai (*colormap*) RGB tertentu. Kedua, setiap *pixel* memiliki format 8 bit sebagai berikut.

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
R	R	R	G	G	G	B	B

Gambar 9. Format 8 bit
Sumber: (Putra, 2010)

Bentuk kedua ini dinamakan 8 bit *truecolor*. Berikut ini adalah warnanya.



Gambar 10. 8 bit *truecolor*
Sumber: (Putra, 2010)



Gambar 11. Contoh citra warna 8 bit dengan palet
Sumber: (Putra, 2010)

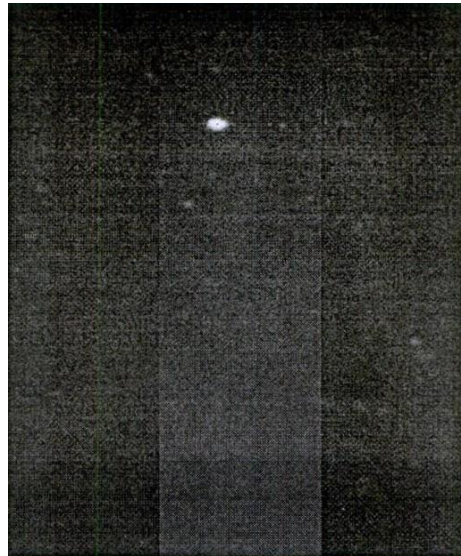
2.2.4 Citra warna (16 bit)

Citra warna 16 bit, biasanya disebut sebagai citra *highcolor*, memiliki setiap *pixel* yang diwakili oleh 2 *byte* memori (16 bit). Warna 16 bit dapat menampilkan 65.536 warna. Dalam formasi bit-nya, komponen merah dan biru masing-masing menggunakan 5 bit di kanan dan kiri, sementara komponen hijau menggunakan 5 bit ditambah 1 bit ekstra. Komponen hijau diberi 6 bit karena penglihatan manusia lebih sensitive terhadap warna hijau.

Bit-15	Bit-14	Bit-13	Bit-12	Bit-11	Bit-10	Bit-9	Bit-8	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
R	R	R	R	R	G	G	G	G	G	G	B	B	B	B	B

Gambar 12. Format 16 bit
Sumber: (Putra, 2010)

Berikut ini adalah deret warna yang dihasilkan.



Gambar 13. 16 bit *highcolor*
Sumber (Putra, 2010)



Gambar 14. Contoh citra warna 16 bit
Sumber: (Putra, 2010)

2.2.5 Citra warna (24 bit)

Setiap *pixel* dalam citra warna 24 bit diwakili oleh 24 bit, memungkinkan total 16.777.216 variasi warna. Variasi ini sudah lebih dari cukup untuk memvisualisasikan semua warna yang dapat dilihat oleh mata manusia, yang dipercaya hanya mampu membedakan hingga 10 juta warna. Setiap komponen warna (RGB) disimpan dalam 1 *byte* data. 8 bit pertama menyimpan nilai biru, 8 bit kedua menyimpan nilai hijau, dan 8 bit terakhir menyimpan nilai merah.



Gambar 15. Contoh citra warna 24 bit
Sumber: (Putra, 2010)

2.3 *Image Resizing*

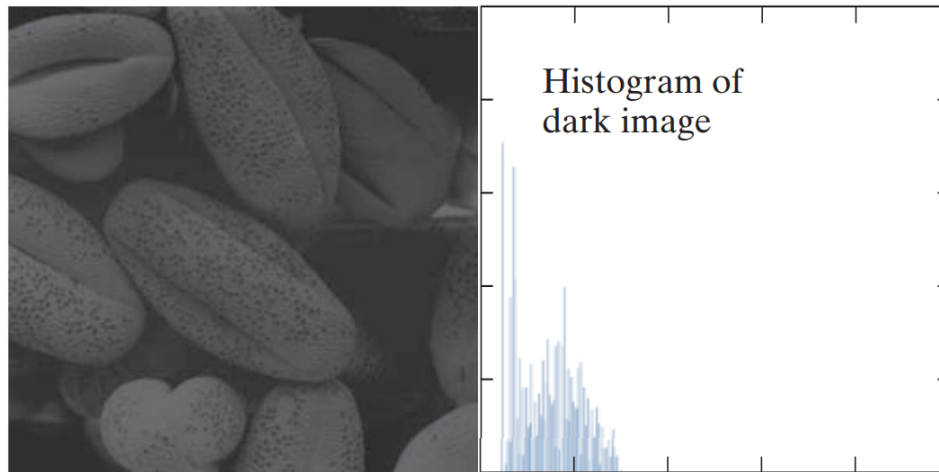
Image resizing adalah proses mengubah ukuran dimensi citra, baik itu memperbesar atau memperkecil ukuran citra, untuk memenuhi kebutuhan tertentu tanpa mengubah struktur atau konten dasar citra tersebut. Dalam *image resizing*, dua metode umum yang sering digunakan adalah *resampling* dan interpolasi. *Resampling* mengubah jumlah *pixel* dalam citra, sedangkan interpolasi menentukan nilai intensitas *pixel* baru saat ukuran citra diubah (Gonzales and Woods, 2018).

2.4 *Histogram Equalization*

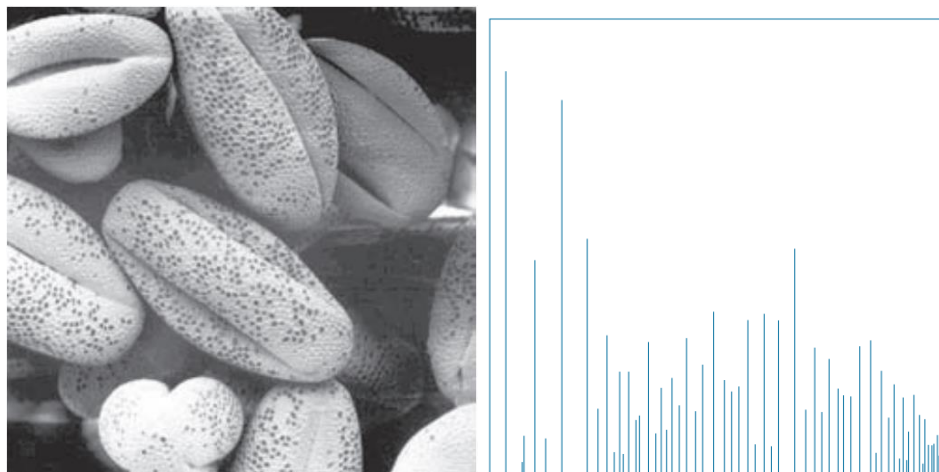
Teknik *histogram equalization* (HE) bertujuan untuk mendistribusikan tingkat keabuan dalam citra sehingga setiap tingkat memiliki probabilitas kemunculan yang sama. *Histogram equalization* meningkatkan kecerahan dan kontras pada citra yang gelap dan berkontras rendah. Pada citra yang gelap, *histogram* biasanya condong ke tingkat keabuan yang lebih rendah, memadatkan informasi citra di bagian gelap (Rahman et al, 2021).

Menurut Gonzalez and Woods dalam “*Digital Image Processing*”, *histogram equalization* adalah salah satu metode yang paling efektif untuk meningkatkan kontras global dalam sebuah citra. Namun, teknik ini mungkin tidak selalu memberikan hasil yang diinginkan jika citra memiliki variasi kontras lokal yang signifikan. Dalam kasus seperti itu, metode lain seperti

Adaptive Histogram Equalization (AHE) atau *Contrast Limited Adaptive Histogram Equalization* (CLAHE) dapat digunakan untuk memperbaiki kontras dengan lebih baik.



(a) citra sebelum diimplementasi.



(b) citra setelah diimplementasi

Gambar 16. Contoh implementasi *histogram equalization*

Sumber: (Gonzalez and Woods, 2018)

Berikut rumus yang digunakan dalam melakukan proses perhitungan *histogram equalization* (Manalu, 2021).

$$K_o = \text{round}\left(\frac{C_i(2^k - 1)}{w \cdot h}\right) \quad (3)$$

dimana,

C_i : Distribusi kumulatif dari nilai skala keabuan ke- i
dari citra asli

round : Fungsi pembulatan kebilangan yang terdekat

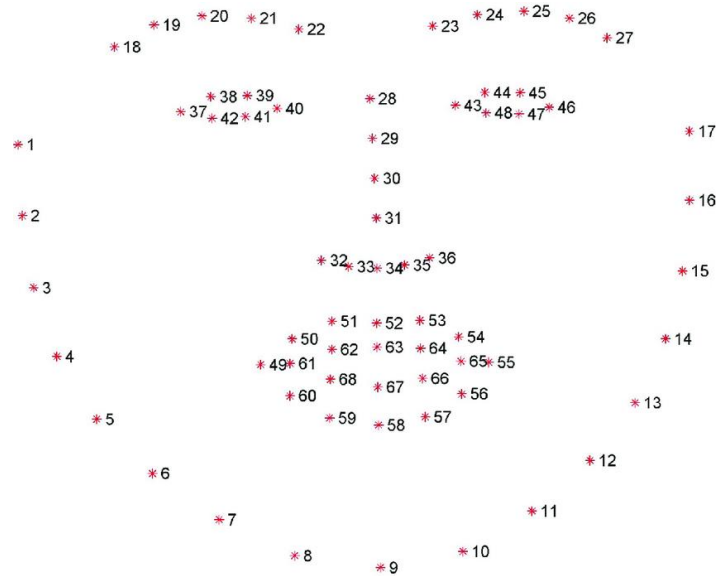
K_o : Nilai keabuan hasil *histogram equaliazation*

w : Lebar citra

h : Tinggi citra

2.5 Dlib

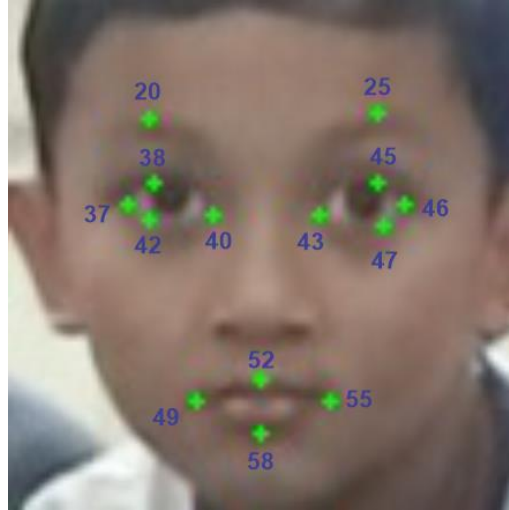
Dlib adalah *library open-source* yang menyediakan *environment* untuk pengembangan perangkat lunak berbasis C++ yang mencakup algoritma *machine learning*, termasuk deteksi wajah manusia dan algoritma *feature point*. Dlib menganalisis bagian-bagian wajah dengan mengekstrak nilai pada wajah yang menghasilkan 128 *dimensional feature vector* (Chen and Sang, 2018). Dlib menerapkan teknik yang disebut *One Millisecond Face Alignment with an Ensemble of Regression Trees*, yang diperkenalkan oleh Kazemi dan Sullivan pada tahun 2014. Dlib digunakan untuk membantu lokalisasi titik-titik penanda pada wajah. Terdapat 68 titik yang telah dilatih dan terintegrasi dalam sebuah file *predictor* dengan ekstensi *.dat* yang disediakan oleh *library dlib*. Titik-titik ini mendeteksi koordinat dari fitur-fitur wajah, sehingga fitur-fitur tersebut dapat diekstraksi sesuai kebutuhan. Berikut adalah indeks dari 68 titik penanda wajah yang direpresentasikan dalam *dataset* iBUG 300-W (Rosiani dkk, 2019).



Gambar 17. Visualisasi 68 koordinat *facial landmark* dari dataset iBUG 300-W
Sumber: (Zein, 2018)

2.6 Geometrical Feature Extraction

Pendekatan berbasis geometri dalam analisis ekspresi wajah melibatkan ekstraksi fitur lokal seperti mulut, alis, dan hidung dari citra wajah. Fitur-fitur ini kemudian digunakan untuk klasifikasi ekspresi wajah. Metode ini bertujuan untuk menangkap deformasi geometris yang terjadi pada komponen wajah ketika seseorang mengekspresikan emosi tertentu (Al-agma et al., 2017). Pada Gambar 17, terdapat 68 *landmark* yang terdeteksi, subset yang relevan dipilih untuk analisis lebih lanjut. Subset ini mencakup titik-titik yang signifikan untuk ekspresi wajah seperti sudut mata, ujung hidung, sudut mulut, dan lain-lain. Hanya 14 *facial landmark* dari 68 yang digunakan untuk mengurangi kompleksitas tanpa mengurangi akurasi (Ashwin et al., 2018).



Gambar 18. *Facial landmarks* dengan 14 titik pada fitur wajah

Terdapat sembilan fitur geometris wajah yang diekstraksi dari 14 titik tersebut. Fitur-fitur tersebut dihitung berdasarkan jarak *euclidean* antara pasangan titik tertentu, yang memberikan informasi tentang bentuk dan ukuran bagian wajah yang berbeda. Notasi $d(p_i, p_j)$ digunakan untuk menyatakan jarak *euclidean* antara dua titik dengan indeks i dan j . Fitur geometris g_i adalah fitur geometris ke- i yang diekstraksi dari jarak antara titik tertentu (Ashwin et al., 2018).

Jarak antara kelopak mata dan alis

$$g_1 = \frac{d(p_{20}, p_{40}) + d(p_{25}, p_{43})}{2} \quad (4)$$

Jarak antara kelopak mata dan bibir

$$g_2 = \frac{d(p_{40}, p_{52}) + d(p_{43}, p_{52})}{2} \quad (5)$$

Lebar mulut

$$g_3 = d(p_{55}, p_{49}) \quad (6)$$

Tinggi mulut

$$g_4 = d(p_{52}, p_{58}) \quad (7)$$

Rasio lebar mulut terhadap tinggi mulut

$$g_5 = \frac{g_3}{g_4} = \frac{d(p_{55}, p_{49})}{d(p_{52}, p_{58})} \quad (8)$$

Jarak antara kedua alis

$$g_6 = d(p_{20}, p_{25}) \quad (9)$$

Jarak antara kelopak mata dari satu mata

$$g_7 = \frac{d(p_{38}, p_{42}) + d(p_{45}, p_{47})}{2} \quad (10)$$

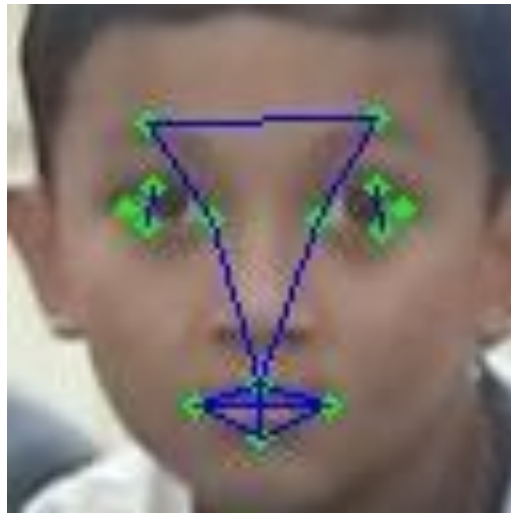
Rata-rata setengah panjang bibir atas

$$g_8 = \frac{d(p_{52}, p_{49}) + d(p_{52}, p_{55})}{2} \quad (11)$$

Rata-rata setengah panjang bibir bawah

$$g_9 = \frac{d(p_{58}, p_{49}) + d(p_{58}, p_{55})}{2} \quad (12)$$

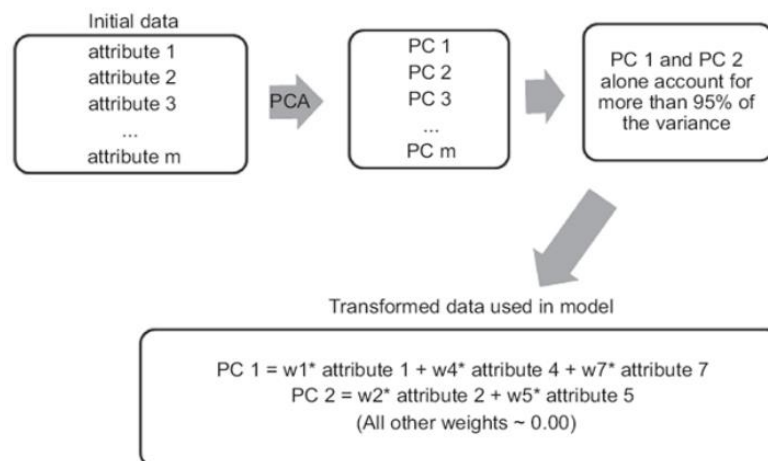
Fitur geometris wajah yang diekstraksi dari 14 titik *landmark* wajah ditampilkan pada Gambar 19 dengan garis biru yang menghubungkan titik-titik *landmark* tersebut.



Gambar 19. Ekstraksi fitur geometri wajah

2.7 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) adalah teknik yang digunakan untuk menyederhanakan kumpulan data yang saling terkait. Proses PCA dimulai dengan standarisasi data yang akan dianalisis. PCA melibatkan perhitungan nilai *eigen* dari matriks kovarian data yang telah distandarisasi. Nilai *eigen* terdiri dari dua komponen utama yaitu *eigenvalue* dan *eigenvector* (Yudistira dkk, 2019). Dengan menggunakan PCA, jumlah variabel awal yang sebanyak n dapat direduksi menjadi k variabel baru yang disebut *principal components*, di mana k lebih sedikit dari n . *Principle components* ini mempertahankan sebagian besar informasi dari variabel asli, sehingga hasil analisis tetap akurat meskipun jumlah variabel berkurang (Nasution, 2019). Skema konseptual pada Gambar 20 menggambarkan bagaimana PCA dapat menyederhanakan dimensi data dari dataset yang memiliki m variabel.



Gambar 20. Kerangka konseptual yang menggambarkan efektivitas penggunaan PCA untuk pemilihan fitur

Sumber: (Kotu & Deshpande, 2015)

Prosedur pengerjaan PCA bertujuan untuk menyederhanakan dan mengeliminasi faktor atau indikator yang kurang dominan dan kurang relevan, tanpa mengurangi makna dan tujuan asli dari data tersebut. PCA dilakukan pada variabel acak x , yang diatur dalam matriks berukuran $n \times n$, dengan setiap baris matriks merepresentasikan observasi dari variabel acak x . Prosedurnya melibatkan langkah-langkah berikut (Nasution, 2019) (Astuti dan Samsuryadi, 2018).

- Normalisasi data:

- Rata-rata

$$\mu_x = \frac{1}{n} \sum_{i=1}^n x_i \quad (13)$$

dimana,

μ_k : Rata-rata dari variabel x

n : Jumlah total observasi

x_i : Rata-rata dari variabel x pada observasi ke- i

- Varians

$$Var(x) = \sigma_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu_x)^2 \quad (14)$$

dimana,

x_i : Nilai observasi ke- i dari variabel x

μ_x : Rata-rata dari data dari variabel x

- Proses normalisasi menggunakan *Z-score*

$$Z_{ij} = \frac{X_{ij} - \mu_j}{\sigma_j} \quad (15)$$

dimana,

Z_{ij} : Nilai *z-score* dari variabel j pada observasi ke- i

X_{ij} : Nilai asli dari variabel j pada observasi ke- i

- Menghitung matriks kovarian

$$Cov(x, y) = \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \mu_{xj})(y_{ij} - \mu_{yj}) \quad (16)$$

dimana,

μ_j dan μ_{yj} : Rata-rata dari data sampel variabel x
dan y

x_i dan y_i : Nilai observasi ke-i dari variabel x

- Mencari *eigenvalues* dan *eigenvector* dari matriks kovarian

$$Det(A - \lambda I) = 0 \quad (17)$$

dimana,

A : Matriks $n \times n$

λ : Nilai *eigenvalue*

I : Matriks identitas (matriks persegi dengan
elemen diagonal utama bernilai 1 sedangkan
elemen lain bernilai 0)

- Menentukan nilai proporsi *principal component*

$$PC(\%) = \frac{\text{Nilai Eigen}}{\text{Varians Kovarian}} \times 100\% \quad (18)$$

- Menghitung bobot faktor berdasarkan *eigenvector*

$$Ax = \lambda x \quad (19)$$

Sehingga diperoleh kombinasi linear:

- $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n$ adalah *eigenvalue* matriks A
- $x_1, x_2, x_3, \dots, x_n$ adalah *eigenvector* yang sesuai dengan *eigenvalue*-nya (λ_n)

Persamaan *eigenvalue* dan *eigenvector* adalah bagian dari *Eigen Value Decomposition* (EVD), dengan persamaan sebagai berikut:

$$A = XDX^{-1} \quad (20)$$

dimana,

A : Matriks $n \times n$ yang memiliki n *eigenvalue* (λ_n)

D : Diagonal matriks yang berisi *eigenvalue*

X : Matriks yang kolom-kolomnya adalah *eigenvector* dari matriks A

X^{-1} : Invers dari matriks *eigenvector* X

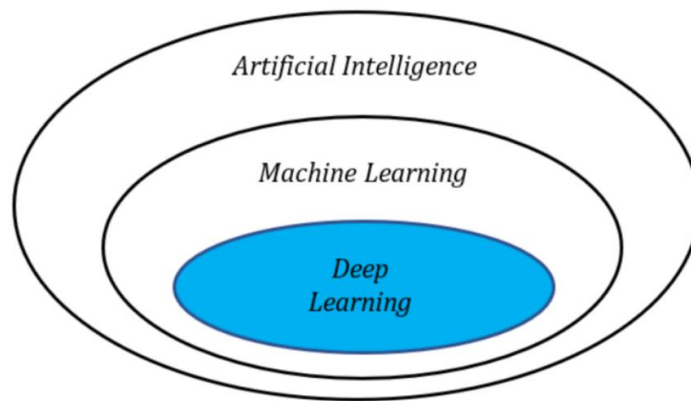
2.8 TensorFlow

Tensorflow adalah *library* perangkat lunak yang fleksibel dan skalabel untuk komputasi numerik menggunakan grafik aliran data. *Library* ini, bersama dengan alat terkait, memungkinkan pengguna untuk memprogramkan dan melatih model *neural network* dan *machine learning* lainnya secara efisien, serta menerapkannya ke dalam produksi. Algoritma inti TensorFlow ditulis dalam bahasa C++ dan CUDA (*Compute Unified Device Architecture*), platform komputasi paralel dan API yang dibuat oleh NVIDIA. TensorFlow menyediakan API dalam beberapa bahasa pemrograman. API Python yang paling lengkap dan stabil. Bahasa lain yang didukung secara resmi termasuk JavaScript, C++, Java, Go, dan Swift. Selain itu, ada paket pihak ketiga yang tersedia untuk bahasa lain seperti C# dan Ruby (Pang et al., 2019).

TensorFlow 2.0 telah mengintegrasikan Keras sebagai antarmuka tingkat tinggi (*high-level API*) untuk membangun dan melatih model *neural network*. Dengan integrasi tersebut, pengguna dapat memanfaatkan kesederhaan dan kemudahan penggunaan Keras. Sehingga memungkinkan pengembangan model yang cepat dan efisien, serta kemampuan untuk memanfaatkan fitur-fitur canggih yang disediakan oleh TensorFlow (Grattarola and Alippi, 2021).

2.9 Deep Learning

Deep learning adalah sub-bidang khusus dari *machine learning* yang menekankan metode pembelajaran representasi data dengan berbagai tingkatan menggunakan modul-modul *non-linear*. Modul-modul ini mengubah representasi dari input mentah menjadi representasi yang lebih abstrak. Inti dari *deep learning* adalah lapisan-lapisannya tidak dirancang oleh manusia, melainkan dipelajari dari data (LeCun et al., 2015).



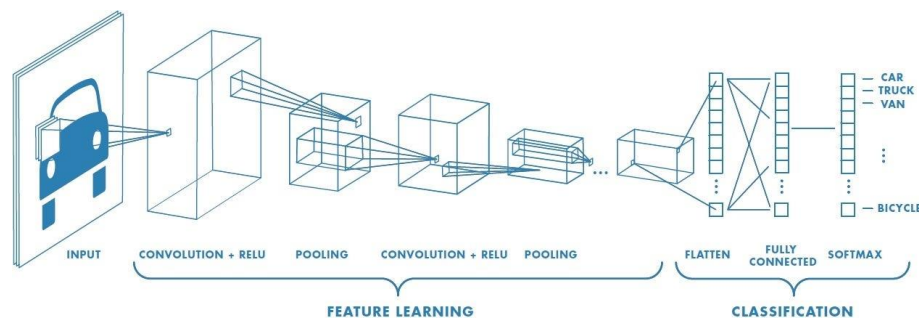
Gambar 21. *Deep learning* sebagai cabang ilmu *artificial intelligence* dan *machine learning*

Sumber: (Heryadi dan Irwansyah, 2020)

Dalam *deep learning*, representasi berlapis ini dipelajari melalui model yang disebut *neural networks*, yang disusun dalam lapisan-lapisan yang saling bertumpuk. Kata “*deep*” mengacu pada banyaknya lapisan dalam model, disebut *depth*. *Deep learning modern* sering melibatkan puluhan hingga ratusan lapisan representasi yang dipelajari secara otomatis dari data pelatihan. Sebaliknya, metode *machine learning* lainnya biasanya hanya mempelajari satu atau dua lapisan representasi, sehingga sering disebut *shallow learning* (Chollet, 2021).

2.10 CNN

CNN adalah pengembangan dari *Multilayer Perceptron* (MLP) dan termasuk dalam kategori *neural network* berjenis *feed forward* (bukan berulang). CNN termasuk dalam jenis *deep neural network* dengan struktur jaringan yang memiliki kedalaman tinggi. CNN banyak digunakan untuk menganalisis citra visual, melakukan deteksi dan pengenalan objek citra (Nugroho dkk, 2020). Metode ini menggunakan proses konvolusi dalam menemukan fitur-fitur pada citra (Alamsyah dan Pratama, 2020). Arsitektur CNN terdiri dari lapis tersembunyi (*Hidden Layer*) yang berisi *convolutional layers*, *pooling layer*, *normalization layers*, *ReLU layer*, *fully connected layer* dan *loss layer* (Alamsyah dan Pratama, 2020).



Gambar 22. Arsitektur CNN

Sumber: (Tinaliah dan Elizabeth, 2021)

2.11.1. Convolutional layer

Semua citra yang melewati lapisan konvolusional akan mengalami proses konvolusi. Lapisan ini akan mengkonversi setiap filter ke seluruh bagian citra masukan dan menghasilkan sebuah *activation map* atau *feature map* 2D. Filter yang terdapat pada *Convolutional Layer* memiliki panjang, tinggi, dan tebal sesuai dengan *channel* data masukan (Nugroho dkk, 2020). Lapisan konvolusi memiliki beberapa parameter penting, yaitu ukuran *kernel*, *skipping factors*, dan *connection table*. *Kernel* dalam *Convolutional Neural Network* (CNN) selalu bergerak melintasi input, sedangkan *skipping factors* menunjukkan jumlah *pixel* yang digeser oleh *kernel* setiap langkah. Ukuran dari *output* pada *feature map* dapat dihitung

berdasarkan parameter-parameter tersebut menggunakan Persamaan 21 (Kusdiananggalih dkk, 2021).

$$M_x^n = \frac{M_x^{n-1} - K_x^n}{S_x^n + 1} + 1; \quad M_y^n = \frac{M_y^{n-1} - K_y^n}{S_y^n + 1} + 1 \quad (21)$$

dimana,

M_x^n, M_y^n : Ukuran *feature maps*

S_x, S_y : *Skipping factors*

K_x, K_y : Ukuran *kernel*

n : Iterasi dalam suatu jaringan konvolusi

<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> </table> Citra	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	1	1	0	0	1	1	0	0	X	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> </table> Kernel	1	0	1	0	1	0	1	0	1
1	1	1	0	0																																
0	1	1	1	0																																
0	0	1	1	1																																
0	0	1	1	0																																
0	1	1	0	0																																
1	0	1																																		
0	1	0																																		
1	0	1																																		
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>4</td><td>3</td><td>4</td></tr> <tr><td>2</td><td>4</td><td>3</td></tr> <tr><td>2</td><td>3</td><td>4</td></tr> </table> <i>Convolved Feature</i>			4	3	4	2	4	3	2	3	4																									
4	3	4																																		
2	4	3																																		
2	3	4																																		

Gambar 23. Contoh konvolusi citra ukuran 5x5x1 dengan 3x3x1 *kernel*

Sumber: (Kusdiananggalih dkk, 2021)

2.11.2. Pooling layer

Pooling layer adalah proses untuk mengurangi ukuran citra, yang bertujuan mengurangi jumlah parameter dan perhitungan dalam jaringan serta membantu mencegah *overfitting*. *Pooling layer* terbagi menjadi dua jenis utama, yaitu *max pooling* dan *average pooling*. *Max pooling* mengambil nilai terbesar dari area yang dipilih, sementara *average pooling* menghitung nilai rata-rata dari area tersebut. Dengan demikian, *max pooling* berfokus pada fitur yang paling dominan dalam area tersebut, sedangkan *average pooling* memberikan gambaran rata-

rata dari fitur dalam area tersebut. Berikut persamaan untuk *max pooling* (Kusdiananggalih dkk, 2021).

$$a_j = \max_{N \times N} a_i^{n \times n} \mu(n, n) \quad (22)$$

dimana,

a_i : Value dari *pooling map*

a_j : Value dari *input map*

$\mu(n, n)$: *Window function*

3	3	2	1	0
0	0	1	3	0
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

Citra

3	3	3
3	3	3
3	2	3

Output

Gambar 24. Contoh *Max pooling layer* 3x3 pada matriks 5x5
Sumber: (Kusdiananggalih dkk, 2021)

2.11.3. *Batch normalization*

Batch normalization bertujuan untuk mengurangi pergeseran kovarian internal, sehingga mempercepat pelatihan jaringan neural. Teknik ini bekerja dengan menormalkan input *layer* berdasarkan *mean* dan *varians* dari *mini-batch*. Proses normalisasi ini menstabilkan distribusi aktivasi di setiap *layer*, sehingga gradien yang dihitung selama *backpropagation* menjadi lebih konsisten. Dengan mengurangi ketergantungan gradien pada skala parameter atau nilai awal, *batch normalization* memungkinkan penggunaan laju pembelajaran yang lebih tinggi tanpa risiko divergensi. Selain itu, *batch normalization* berfungsi sebagai bentuk regulasi yang dapat mengurangi *overfitting*, sehingga sering kali mengurangi kebutuhan untuk teknik regulasi tambahan seperti *dropout* (Loffe and Szegedy, 2015).

- Menghitung *mean* dan *variance*

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (23)$$

dimana,

μ_B : *Mean* dari aktivasi dalam *mini-batch*

m : Jumlah sampel dalam *mini-batch*

x_i : Aktivasi dari neuron i dalam *mini-batch*

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (24)$$

dimana,

σ_B^2 : *Variance* dari aktivasi dalam *mini-batch*

- Normalisasi aktivasi

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 - \epsilon}} \quad (25)$$

dimana,

$\hat{x}_i$: Nilai aktivasi yang telah dinormalisasi

ϵ : Nilai kecil untuk mencegah pembagian dengan nol (biasanya 10^{-7})

- Transformasi linier

$$y_i = \gamma \hat{x}_i + \beta \quad (26)$$

dimana,

y_i : Nilai aktivasi yang telah dinormalisasi dan ditransformasi

γ : Parameter *trainable* untuk skala

β : Parameter *trainable* untuk pergeseran

2.11.4. Fungsi aktivasi

Fungsi aktivasi adalah fungsi *non-linear* yang memungkinkan jaringan *neural* untuk mentransformasi citra input ke dimensi yang lebih tinggi, sehingga memungkinkan klasifikasi yang lebih efektif (Tinaliah dan Elizabeth, 2021). Dalam *Convolutional Neural Network* (CNN), terdapat beberapa jenis fungsi aktivasi yang umum digunakan yaitu fungsi aktivasi *Rectified Linear Unit* (ReLU) dan *softmax*.

a. Fungsi aktivasi ReLU

ReLU memiliki kemampuan untuk memproses citra dalam jumlah besar dengan cepat dan efisien. Fungsi ReLU mentransformasi nilai input menjadi nilai positifnya, atau nol jika nilai input negatif, sehingga memperkenalkan *non-linear* ke dalam model tanpa memperlambat komputasi. Berikut persamaan fungsi ReLU (Kusdiananggalih dkk, 2021).

$$f = \max(0, x) \quad (27)$$

b. Fungsi aktivasi softmax

Softmax adalah salah satu fungsi aktivasi yang digunakan dalam jaringan *neural*, terutama untuk tugas klasifikasi multi-kelas. Fungsi ini mengubah output dari model menjadi probabilitas, memastikan bahwa total probabilitas dari semua kelas adalah 1. Persamaan fungsi *softmax* adalah sebagai berikut (Kusdiananggalih dkk, 2021).

$$f_i(x) = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}} \quad (28)$$

where $i = 1, 2, 3, \dots, k$

$$0 \leq f_i(x) \leq 1$$

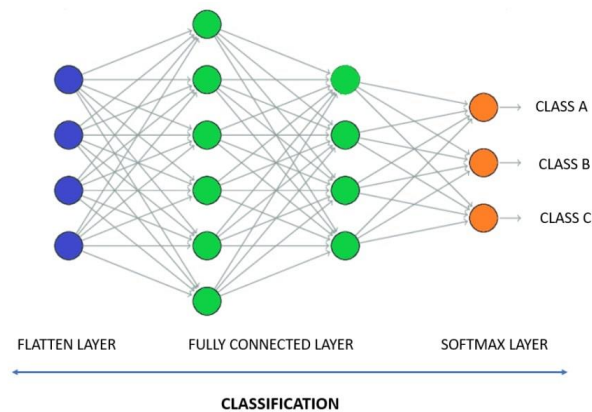
dimana,

k : Jumlah input

x : Vektor input

2.11.5. Fully connected layer

Feature map yang dihasilkan tahap sebelumnya berbentuk *multidimensional array*. Sebelum masuk ke tahap *fully connected layer*, *feature map* tersebut harus melalui proses “*flatten*” atau *reshape*. Proses *flatten* mengubah *multidimensional array* menjadi vektor satu dimensi yang akan digunakan sebagai input pada *fully connected layer*. *Fully connected layer* terdiri dari beberapa komponen, seperti *hidden layer*, *activation function*, *output layer*, dan *loss function* (Nugroho dkk, 2020).



Gambar 25. Contoh *fully connected layer*

2.11.6. Loss layer

Lapisan ini adalah lapisan akhir dalam *Convolutional Neural Network* (CNN) yang bertugas menentukan *penalty* untuk hasil yang tidak sesuai dengan label atau target selama proses pelatihan (penentuan bobot dan bias). Beberapa fungsi umum yang digunakan dalam lapisan ini meliputi *sigmoid cross-entropy loss*, *softmax loss*, *Euclidean loss*, dan sebagainya (Alamsyah dan Pratama, 2020).

2.11 Confusion Matrix

Perhitungan nilai akurasi dalam pembelajaran mesin sering dilakukan menggunakan *confusion matrix*. Pada dasarnya, *confusion matrix* memberikan perbandingan antara hasil klasifikasi yang dilakukan oleh model dengan hasil klasifikasi yang sebenarnya (Gunawan dkk, 2021).

Tabel 1. *Confusion matrix* dari suatu klasifikasi

Predicted Values	Actual Values	
	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Sumber: (Anggraini, 2020)

Penjelasan:

- TP (*True Positive*): jumlah prediksi yang benar dalam kelas positif
- FP (*False Positive*): jumlah prediksi yang salah dalam kelas positif
- FN (*False Negative*): jumlah prediksi yang benar dalam kelas positif
- TN (*True Negative*): jumlah prediksi yang benar dalam kelas positif

Dalam *confusion matrix*, terdapat beberapa metrik performa yang umum digunakan, yaitu *accuracy*, *precision*, dan *recall*. Berikut adalah persamaan untuk menghitung nilai *accuracy*, *precision*, dan *recall* (Gunawan, dkk, 2021). Berikut persamaan untuk menghitung nilai *accuracy*, *precision*, dan *recall*:

- Accuracy* (akurasi): mengukur proporsi total prediksi yang benar dari keseluruhan prediksi.

$$Accuracy = \frac{TP + TN}{N} \quad (29)$$

dimana,

 N : Jumlah total data sampel

- Precision* (presisi): mmengukur proporsi prediksi positif yang benar dari semua prediksi positif

$$Precision = \frac{TP}{TP + FP} \quad (30)$$

- Recall* (sensitivitas): mengukur proposi kondisi positif yang benar-benar terdeteksi oleh model.

$$Recall = \frac{TP}{TP + FN} \quad (31)$$

- d. *F1-Score*: mempertimbangkan metrik presisi dan *recall* secara bersamaan dan memberikan keseimbangan antara keduanya.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (32)$$