

BAB 2

TINJAUAN PUSTAKA

2.1 Kecerdasan Buatan

Kecerdasan buatan atau biasa disebut juga dengan *artificial intelligence* (AI) merupakan teknologi di bidang ilmu komputer yang menjelaskan tentang kecerdasan manusia ke dalam mesin (komputer) untuk menyelesaikan berbagai persoalan dan pekerjaan seperti yang dilakukan oleh manusia bahkan lebih baik dari manusia. Kecerdasan buatan adalah kecerdasan yang ditambahkan oleh manusia ke dalam suatu sistem teknologi, diatur dan dikembangkan dalam konteks ilmiah, bentukan dari kecerdasan entitas ilmiah yang ada.

Pada dasarnya kecerdasan buatan atau *artificial intelligence* (AI) adalah suatu pengetahuan yang membuat komputer dapat meniru kecerdasan manusia sehingga komputer dapat melakukan hal-hal yang dikerjakan oleh manusia di mana membutuhkan suatu kecerdasan, contohnya melakukan analisis penalaran untuk mengambil kesimpulan atau keputusan. (Subakti, H., Romli, I., Nur Syamsiyah, S. T., Budiman, A. A., Kom, M., Herianto, S. P., & MSI, M. 2022).

2.2 Face Recognition

Face recognition adalah teknik biometrik yang memungkinkan komputer atau mesin untuk mengenali wajah manusia. Teknologi pengenalan wajah dapat diterapkan dalam kehidupan sehari-hari untuk mempermudah aktivitas manusia. Dalam implementasi pengenalan wajah, banyak metode yang dapat digunakan, salah satunya adalah *machine learning*, yang mana hasil tangkapan dari kamera akan dilakukan pencocokan dengan menggunakan data yang telah dilatih sebelumnya menggunakan teknologi *machine learning*. Untuk dapat membaca dan memproses data inputan berupa citra, dapat menggunakan *library* OpenCV (Prapdipta, Darlis dan Rangkuti. 2020).

2.3 Computer Vision

Computer vision merupakan cabang dari *artificial intelligence* (AI) atau kecerdasan buatan yang mempelajari ilmu mengenai bagaimana personal komputer bisa mengenali objek yang diamati. Implementasi dari teknologi AI sudah banyak

sekali digunakan, baik pada teknologi *smartphone* juga pada dunia robotika. *computer vision* juga memungkinkan komputer bisa melihat objek atau benda yang terdapat pada lingkungan sekitar. Maka komputer bisa menganalisis benda atau gambar yang terdapat pada depannya sehingga informasi yang sudah terdeteksi bisa diterima & mampu membentuk perintah tertentu (Santoso, B. 2021).

2.4 Machine Learning

Machine Learning merupakan bidang studi yang berfokus pada desain dan analisis algoritma untuk melakukan komputer dapat belajar tanpa diprogram secara eksplisit. *Machine Learning* berisi algoritma umum yang bersifat umum dimana algoritma tersebut dapat menghasilkan hal-hal yang berguna dari kumpulan data tertentu tanpa harus menulis kode yang spesifik. Pada intinya adalah algoritma yang umum tersebut ketika diberikan sejumlah data maka akan dibangun sebuah sistem atau model yang berasal dari data tersebut. Sebagai contohnya adalah sebuah algoritma untuk mengenali tulisan tangan dapat digunakan untuk dapat mendeteksi email yang berisi spam dan bukan spam tanpa harus mengganti kode. Algoritma yang sama ketika diberikan data pelatihan yang berbeda akan menghasilkan logika klasifikasi yang berbeda. : (MACHINE LEARNING : Teori, Studi Kasus dan Implementasi Menggunakan Python, 2021).

2.4.1 Supervised Learning

Supervised learning adalah metode pembelajaran klasifikasi yang terarah dengan proses pembelajaran dimana seluruh kumpulan data akan diberikan label. Dalam *Supervised learning* model dapat dilatih dalam dataset *training* dan diawasi untuk membuat klasifikasi atau prediksi sesuai dengan *output* berupa data berlabel yang ditentukan sebelumnya berdasarkan pola data *training* yang ada (Hussein Saddam, 2022).

2.4.2 Unsupervised Learning

Unsupervised Learning merupakan kebalikannya dimana proses pembelajaran dapat dilakukan tanpa adanya petunjuk atau arahan. Algoritma pada komputer yang bekerja untuk menemukan pola di dalam data secara matematis, metode *unsupervised learning* terjadi ketika memiliki sejumlah data masukan (X) dan tanpa variabel *output* yang berhubungan. Dengan

pembelajaran yang menggunakan algoritma machine learning ini dapat melakukan analisa dan mengklasifikasikan kumpulan data yang tidak berlabel. Algoritma ini disebut tanpa pengawasan karena menemukan pola tersembunyi dalam data tanpa perlu campur tangan manusia (Hussein Saddam, 2022).

2.4.3 Reinforcement Learning

Reinforcement learning merupakan sebuah komputer yang akan berinteraksi dengan sebuah lingkungan yang sangat dinamis dimana komputer dapat melakukan tugas tertentu. Melalui sebuah algoritma, mesin akan mempelajari bagaimana cara membuat keputusan yang spesifik berdasarkan lingkungan yang berubah-ubah. Selain itu cara belajar tersebut, *machine learning* dapat juga dikategorikan berdasarkan proses pembelajaran yang akan dilakukan secara bertahap (*Batch Learning*) atau secara langsung (*on the fly – Online Learning*). Dasar dari *online learning* dapat menghasilkan suatu model yang dapat melakukan proses pembelajaran data baru secara *realtime* atau mendekati *realtime*. Kemudian sedangkan *Batch Learning*, *data training* akan di bagi-bagi menjadi beberapa bagian lalu setiap bagiannya akan dipelajari secara terpisah pada waktu yang berbeda (MACHINE LEARNING : Teori, Studi Kasus dan Implementasi Menggunakan Python, 2021).

2.5 Pengolahan Citra Digital

Pengolahan citra digital secara umum menunjuk pada proses komputer pada citra dua dimensi. Bahkan disebutkan Anil K Jain (Jain, 1989) dalam pengertian yang luas pengelolaan citra digital merupakan implementasi dari suatu proses digital pada data dua dimensi. Dalam sebuah citra yang mengalami penurunan kualitas karena mengandung *noise*, *blurring*, kontras, kurang tajam, buruk, dan lainnya akan sulit di interpretasikan baik oleh manusia ataupun oleh komputer. Sehingga pada citra itu perlu dilakukan sejumlah pemrosesan dan manipulasi agar dapat menghasilkan citra dengan kualitas informasi yang lebih baik, sehingga dapat mudah di interpretasi.

Operasi pengolahan citra diterapkan pada citra bila:

- Modifikasi citra perlu dilakukan agar meningkatkan kualitas penampakan atau untuk melihatkan beberapa aspek informasi yang ada dalam citra (*image*

enhancement). Operasi ini memiliki tujuan memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra. Contohnya dapat dilakukan dengan perbaikan kontras, penajaman dan lainnya.

- Terdapat cacat pada citra sehingga perlu dihilangkan (*image retoration*). Contohnya dapat dilakukan dengan cara penghapusan noise, *debluring* (penghilangan kesamaran) karena citra terlihat kabur.
- Elemen dalam citra harus dikelompokkan, dicocokkan dan diukur (*segmentation*). Operasi ini berkaitan erat dengan (*pattern recognition*) pengenalan pola.
- Perlu ekstraksi ciri-ciri tertentu (contoh: tekstur, warna, bentuk) dalam pengidentifikasian objek (*image analysis*).
- Sebagian citra perlu digabung dengan citra lain (*image recontruction*). contohnya adalah beberapa foto rontgen digunakan untuk membentuk ulang gambar organ tubuh.
- Menyembunyikan informasi rahasia (berupa teks/citra) dalam citra (*streganografi*).

Aspek dalam pengolahan citra digital meliputi peningkatan kualitas citra, restorasi citra, segmentasi citra. Secara bururutan peningkatan kualitas citra. Yang dilakukan dalam pengolahan citra digital yaitu:

- Akuisisi

Proses akuisisi citra adalah pemetaan suatu pandangan menjadi citra kontinu dengan menggunakan sensor. Contohnya kita dapat membuat sebuah citra digital dengan kamera atau scanner.

- Preprocessing

Pada praproses ini bertujuan untuk meningkatkan kualitas kecerahan dan kontras, penajaman suatu citra, menghilangkan *noise*, menghilangkan gangguan geometrik/radiometrik.

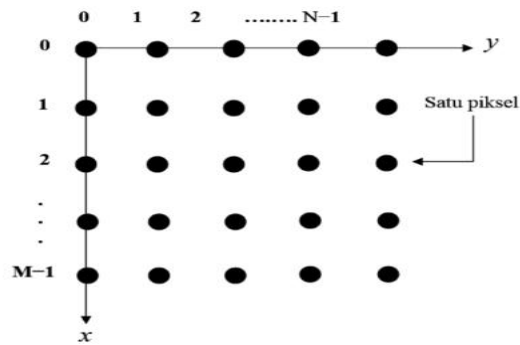
- Segmentasi

Melakukan partisi citra menjadi *region of interest* (ROI) atau wilayah- wilayah objek (*internal properties*), menentukan garis batas wilayah objek (*external shape characteristics*).

Definisi citra adalah sesuatu yang merepresentasikan objek atau beberapa objek dalam bidang dua dimensi. Atau juga bisa dikatakan citra adalah visualisasi, kemiripan dari suatu objek. Citra dapat dikategorikan kedalam citra kontinu dan citra digital. Citra kontinu dihasilkan dari sistem optik yang menerima sinyal analog. Citra diskrit/digital dihasilkan melalui proses digitalisasi terhadap citra kontinu. Suatu citra didefinisikan sebagai $f(x,y)$ memiliki ukuran M baris dan N kolom dengan x dan y merupakan koordinat spasial, dan amplitud f dititik koordinat (x,y) sebagai intensitas (skala keabuan). Disebut sebagai citra digital jika nilai (x,y) dan nilai amplitud f secara menyeluruh (*finite*) dan bernilai diskrit.

Citra digital adalah citra yang dapat diolah dengan komputer. Lalu sedangkan citra dihasilkan dari peralatan digital (citra digital) langsung dapat diolah dengan komputer. Karena penyebabnya adalah di dalam peralatan digital terdapat sistem sampling dan kuantisasi. Sedangkan peralatan analog tidak dilengkapi kedua sistem tersebut. Sistem sampling merupakan sistem yang dapat mengubah citra kontinu menjadi citra digital dengan cara membagi citra analog menjadi M baris dan N kolom, sehingga menjadi citra diskrit. Semakin besar nilai M dan N , semakin halus juga citra digital yang akan dihasilkan. Pertemuan antara baris dan kolom itu disebut dengan piksel. Sistem kuantisasi adalah sistem yang melakukan pengubahan intensitas analog ke intensitas diskrit, sehingga dengan proses tersebut dilakukan untuk membuat gradasi warna sesuai dengan kebutuhan. Maka kedua sistem inilah yang bertugas untuk memotong-motong citra menjadi M baris dan N kolom (proses sampling) sekaligus menentukan besarnya intensitas yang terdapat pada titik tersebut (proses kuantisasi), sehingga dapat menghasilkan resolusi citra yang akan diinginkan.

Hasil sampling dan kuantisasi dari sebuah citra adalah bilangan real yang membentuk sebuah matriks M baris dan N kolom. Maka ukuran citra adalah $M \times N$. Secara umum, sistem koordinat yang dipergunakan untuk mewakili citra dalam teori pengolahan citra seperti digambarkan pada gambar 2.1. Yaitu seperti citra digital diwakili oleh matriks yang terdiri dari M baris dan N kolom, di mana perpotongan antara baris dan kolom disebut piksel. Maka piksel memiliki dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang ada pada koordinat (x,y) adalah $f(x,y)$, yaitu besar intensitas atau warna dari piksel di titik itu.



Gambar 2. 1 Sistem koordinat yang digunakan mewakili citra

Maka, Sebuah citra digital dapat juga ditulis dalam bentuk matriks berikut ini pada gambar 2.2 yaitu :

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

Gambar 2. 2 Matriks Citra Digital $f(x,y)$

Keterangan :

M = Jumlah *pixel* baris (*row*)

N = Jumlah *pixel* kolom (*column*)

Berdasarkan gambar 2.1, maka secara matematis citra digital dapat dituliskan sebagai fungsi intensitas $f(x,y)$, dimana harga x (baris) dan y (kolom) merupakan koordinat posisi dan $f(x,y)$ merupakan nilai fungsi pada setiap titik (x,y) yang mempunyai besar intensitas citra atau tingkat keabuan atau warna dari piksel titik tersebut (Zonyfar, C. 2020).

2.5.1 Jenis Citra Digital

Secara umum, citra digital dibagi menjadi tiga jenis: citra berskala keabuan, citra biner dan citra berwarna.

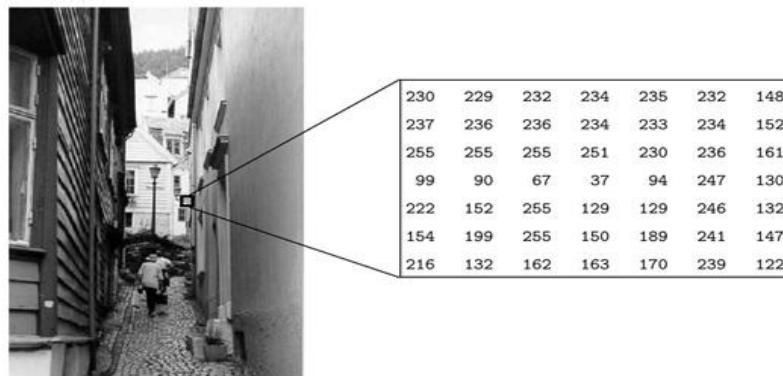
a. Citra Berskala Keabuan

Citra berskala keabuan atau biasa disebut dengan citra *grayscale* merupakan citra yang hanya mempunyai jumlah kanal satu yang ditampilkan hanyalah nilai intensitasnya saja. Satu kanal setiap piksel berwarna abu-abu, biasanya dari 0 (hitam) hingga 255 (putih). Dalam rentang tersebut setiap piksel diwakili oleh 1 byte (8 bit). Sehingga citra ini sering juga dikenal sebagai citra 8 bit, citra *intensity*, *grayscale*, atau *gray level*. Cara mengubah citra berwarna RGB kedalam citra berskala keabuan dapat dilakukan dengan cara merata-ratakan semua nilai piksel pada RGB. Seperti pada persamaan sebagai berikut ini :

$$y = \frac{1}{3}(R + G + B) \quad (2.1)$$

Sedangkan pada MATLAB untuk dapat mengubah citra berwarna menjadi RGB kedalam warna grayscale sesuai dengan persamaan berikut :

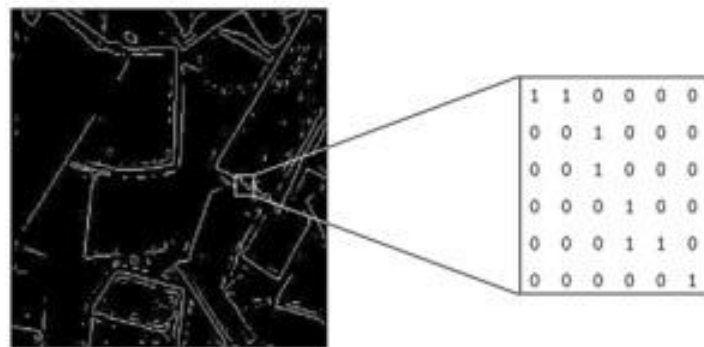
$$Grayscale = 0.299R + 0.587G + 0.144B \quad (2.2)$$



Gambar 2. 3 Citra berskala keabuan

b. Citra Biner

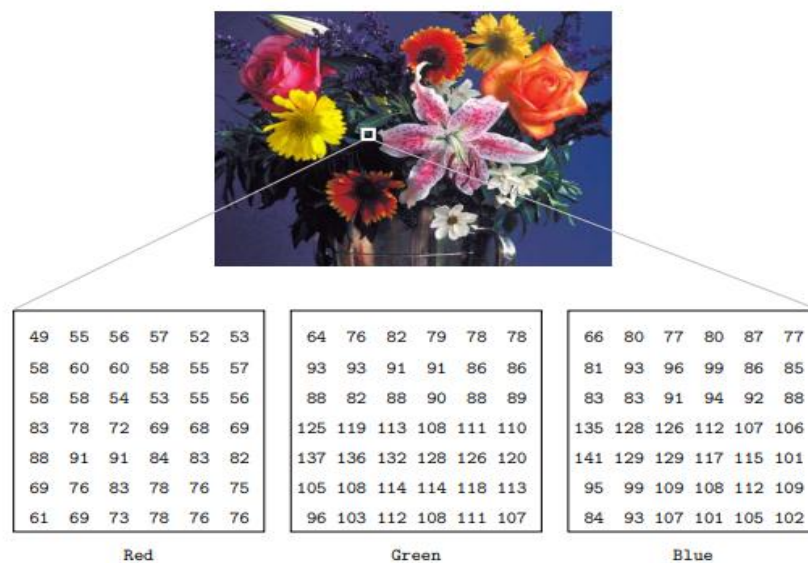
Citra biner adalah citra yang mana setiap pikselnya hanya mempunyai hitam atau putih. Citra biner merupakan citra yang efisien dalam penyimpanan, dikarenakan hanya membutuhkan satu bit saja per piksel. Jenis citra ini sesuai untuk merepresentasikan teks (cetak atau tulis tangan). Citra biner juga sering berfungsi sebagai *masking* atau proses segmentasi. Agar dapatkan citra biner membutuhkan citra berskala keabuan yang kemudian dilakukan *thresholding*. Apabila nilai piksel pada citra berskala keabuan lebih besar atau sama dengan nilai ambang batas, maka piksel tersebut dapat di ubah menjadi 1.



Gambar 2. 4 Binary Image

c. Citra Berwarna

Citra berwarna merupakan citra yang memiliki 3 kanal warna didalamnya. Citra berwarna (*truecolor*) juga sering disebut dengan citra RGB, karena dimodelkan kedalam ruang warna dengan R (*red*/merah), G (*green*/hijau), B (*blue*/biru) sebagai pembentuk komponennya. RGB terdapat standar yang digunakan untuk menampilkan citra yang berwarna pada layer komputer, walaupun seperti itu terdapat juga citra berwarna yang menggunakan warna lainnya, seperti ruang warna HSV, CMYK, dan Lab (Zonyfar, C. 2020).



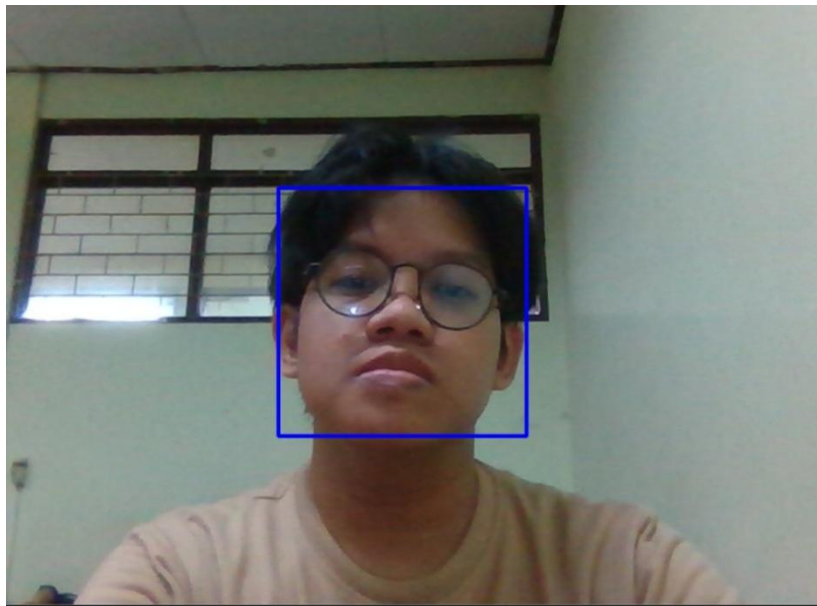
Gambar 2. 5 RGB Image

2.6 Haar Cascade Classifier

Algoritma *Haar Cascade Classifier* merupakan salah satu *library* yang tersedia dalam OpenCV, dibangun dengan menggunakan bahasa C/C++ dengan *API (Application Programming Interface)* python yang dapat digunakan untuk mendeteksi objek pada image digital. Algoritma ini dapat dengan cepat mengenali objek, termasuk wajah manusia, secara *real time*. Algoritma klasifikasi *Haar Cascade* memiliki keunggulan lebih cepat karena hanya bergantung pada jumlah piksel kuadrat pada citra (Abidin, S, 2018).

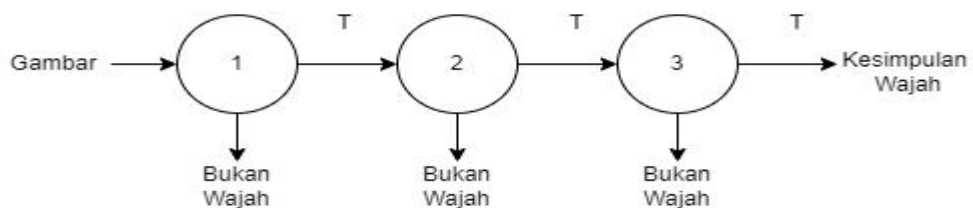
Untuk dapat memproses pendeteksian wajah digunakan algoritma *haar cascade classifier* digunakan untuk mengidentifikasi objek dalam *image digital*. Istilah *haar* berarti menunjukkan suatu fungsi matematika (*Haar Wavelet*) yang berbentuk kotak. Awalnya, pemrosesan gambar hanya terdiri dari pengecekan nilai RGB setiap piksel, tetapi metode ini terbukti tidak efektif. Kemudian Viola dan Jones mengembangkannya menjadi fitur *haar-like feature*. Fungsi *haar-like feature* dapat memproses gambar dalam kotak-kotak yang memiliki banyak piksel di dalam kotak tersebut. kemudian kotak tersebut diproses dan dapat menghasilkan nilai yang berbeda yang menunjukkan area putih dan hitam. Sehingga nilai-nilai ini digunakan sebagai dasar untuk pemrosesan gambar.

Haar like feature memproses citra dalam sebuah kotak persegi dengan menggunakan ukuran tertentu misalnya seperti pada gambar 2.6



Gambar 2. 6 Fitur pada Haar-Like Feature

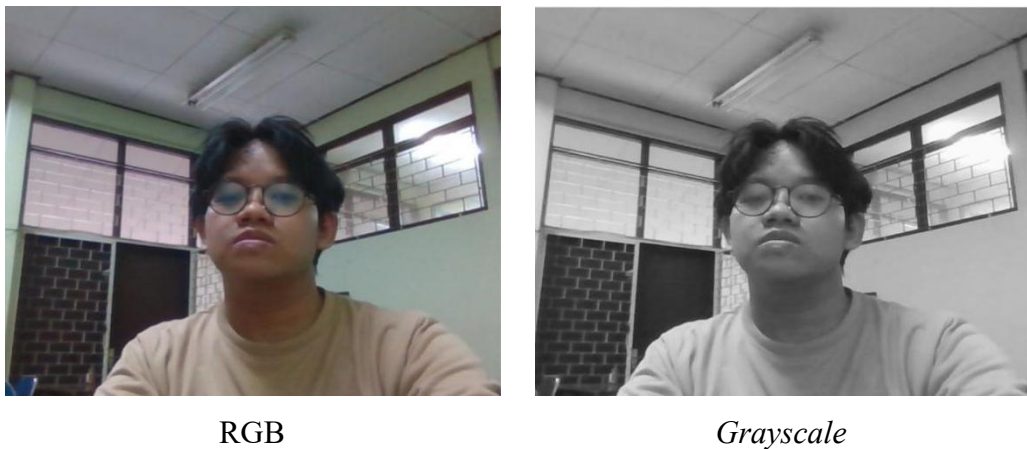
Pada kotak biru pada gambar 2.6 tersebut terjadi proses *filtering* atau penyaringan untuk menentukan ada atau tidaknya suatu objek yang akan dideteksi. Proses tersebut akan dilakukan secara kompleks dan bertingkat. Seperti yang akan ditunjukkan pada gambar filter berikut ini



Gambar 2. 7 Alur metode haar cascade classifier

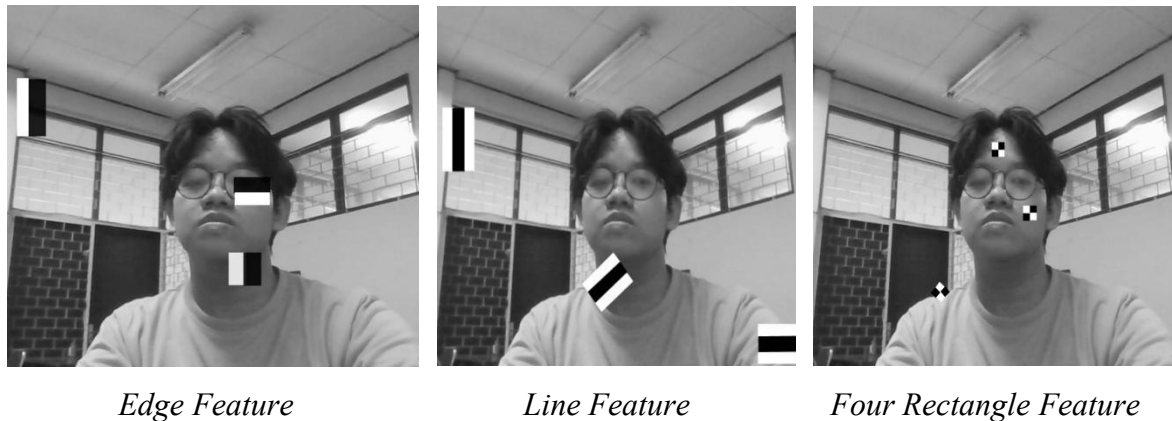
2.6.1 Haar Feature

Haar Feature merupakan fitur yang mempunyai dasar pada *Wavelet Haar*. *Wavelet Haar* adalah gelombang bujur sangkar (1 interval rendah dan interval tinggi). Untuk dua dimensi, yaitu berisi hitam dan putih. Proses pertama yang dilakukan oleh metode *Haar Cascade Classifier* adalah dengan merubah image RGB menjadi citra *grayscale*. Berikut adalah contoh *haar-like feature* dapat dilihat pada gambar 2.8.



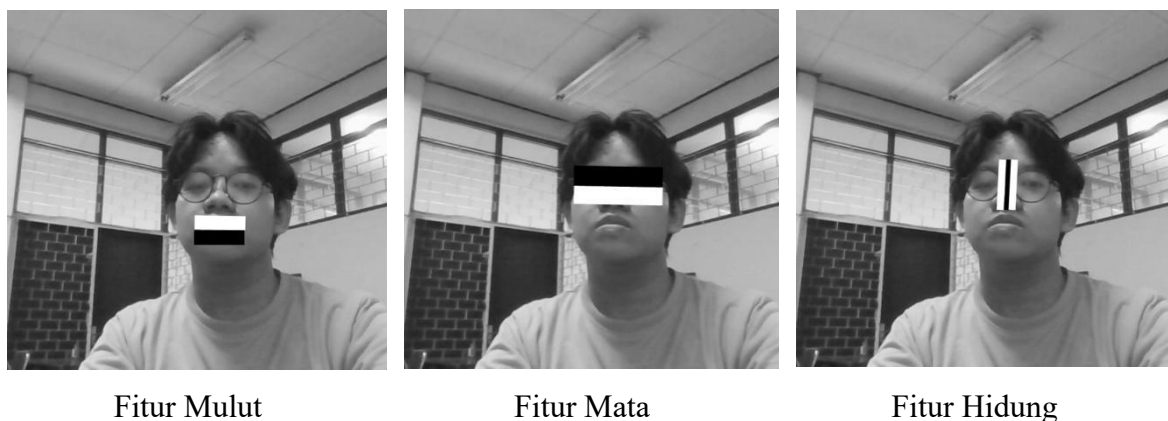
Gambar 2. 8 Perubahan citra RGB menjadi citra Grayscale

Setelah citra *image* dikonversi menjadi citra *grayscale*, proses selanjutnya yaitu memilih fitur Haar yang ada dalam algoritma *Haar Cascade Classifier* disebut dengan *haar-like feature*. Dalam algoritma *Haar Cascade Classifier* Ada beberapa jenis fitur yang bisa digunakan seperti *Edge-feature*, *Line feature*, dan *Four-Rectangle feature*. Pada fitur tersebut digunakan untuk mencari fitur wajah seperti mata, hidung dan mulut. Pada setiap kotak-kotak fitur tersebut terdiri dari beberapa piksel dan akan dihitung selisih antara nilai piksel pada kotak putih dengan nilai piksel pada kotak hitam. Maka apabila nilai selisih antara daerah putih dengan daerah hitam di atas nilai ambang (*threshold*), maka daerah tersebut dinyatakan memiliki fitur. Proses pemilihan fitur dapat dilihat pada gambar 2.9.



Gambar 2. 9 Macam-macam variasi feature pada haar

Untuk dapat memilih fitur mata, hidung, dan mulut maka digunakan kotak-kotak fitur yang bisa dilihat pada gambar 2.10 berikut ini.



Gambar 2. 10 Fitur pada mulut, mata dan hidung.

Terdapat fitur *Haar* ditentukan dengan mengurangi rata-rata piksel pada daerah hitam dari rata-rata daerah yang putih. Jika terdapat nilai perbedaannya diatas nilai ambang atau *threshold*, maka dapat dikatakan bahwa terdapat fitur tersebut . Nilai *Haar-like feature* adalah perbedaan dari jumlah nilai-nilai piksel *gray level* dalam kotak hitam dan kotak putih (Munawir, M., Fitria, L., & Hermansyah, M. 2020) .

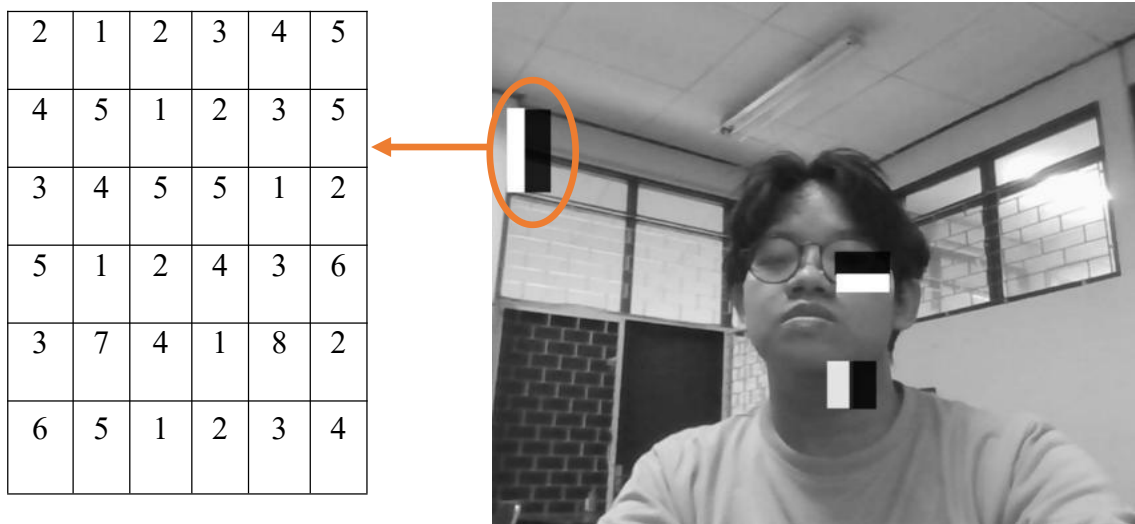
Pada umumnya citra wajah yang menghadap ke depan daerah mata, tepi hidung, mulut dan dagu cenderung lebih gelap dibandingkan dengan daerah kedua pipi, dagu dan kening untuk mempermudah dan mempercepat proses perhitungan nilai *Haar* pada

sebuah *image*, *Algoritma Haar cascade Classifier* memiliki sebuah perhitungan yang disebut dengan *integral image* (Prathivi, R., & Kurniawati, Y. 2020).

2.6.2 Integral Image

Integral Image merupakan suatu teknik penjumlahan intensitas piksel pada suatu daerah yang menghasilkan representasi baru dari citra sebelumnya. *Integral image* sering digunakan pada algoritma untuk pendeteksian wajah dengan menggunakan *integral image* proses perhitungan dapat dilakukan hanya dengan satu kali scan saja dan membutuhkan waktu yang sangat cepat dan akurat. *Integral image* digunakan untuk menghitung hasil penjumlahan nilai piksel pada daerah yang dideteksi oleh *fitur haar*.

Nilai-nilai piksel yang akan dihitung merupakan nilai-nilai piksel dari sebuah citra masukan yang dilalui oleh fitur *haar* pada saat pencarian fitur wajah. Setiap jenis fitur yang akan digunakan pada setiap kotak-kotaknya terdiri dari beberapa piksel. Jadi apabila ada sebuah citra masukan yang dilalui oleh fitur *haar* dapat dilihat pada gambar 2.11



Gambar 2. 11 Nilai piksel pada sebuah fitur

Dapat dilihat pada gambar 2.11 terdapat sebuah feature haar pada gambar yang telah dikonversi ke *grayscale*, maka untuk dapat menentukan nilai fitur tersebut dapat digunakan pada persamaan rumus 2.3 sebagai berikut .

$$\text{NILAI FITUR} = |(\text{totalapikselahitam}) - (\text{total piksel putih})| \quad (2.3)$$

Ada berbagai cara untuk mengetahui nilai dari piksel tersebut, yaitu bisa dilihat pada persamaan rumus 2.2. dan juga diperlukan Sebuah teknik yang dapat dilakukan dengan cepat sehingga dapat diterapkan pada ratusan fitur *haar* pada berbagai skala. Teknik yang efisien untuk ini adalah *integral image*. Maka perhitungan tersebut dapat ditulis dengan rumus *integral image* sebagai berikut (Munawir, M., Fitria, L., & Hermansyah, M. 2020).

$$s(x, y) = i(x, y) + s(x - 1, y) + s(x, y - 1) - s(x - 1, y - 1) \quad (2.4)$$

Keterangan :

$s(x, y)$: Merupakan nilai citra integral pada bidang (x, y)

$i(x, y)$: Intensitas/nilai piksel citra asli.

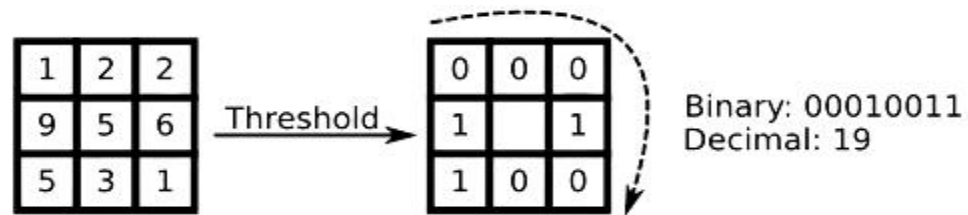
$s(x-1, y)$: Merupakan nilai citra integral dari piksel tetangga sebelah kiri.

$s(x, y-1)$: Merupakan nilai citra integral dari piksel tatangga sebelah atas.

$s(x-1, y-1)$: Merupakan nilai citra integral diagonal dari $s(x, y)$

2.7 Local Binary Pattern Histogram (LBPH)

Local Binary Pattern (LBP) adalah salah satu dari metode yang terkenal dalam mengenali sebuah objek. Sederhana tetapi sangat efisien yang dapat melabeli *pixel* berdasarkan suatu gambar dengan menggunakan ambang batas lingkungan setiap *pixel* dan hasilnya sebagai bilangan biner. Dengan menggabungkan LBP dan *Histogram* bisa didapatkan deskripsi fitur yang bisa dipakai untuk merepresentasikan gambar wajah. Salah satu metode pendeteksian objek yang paling umum, Dalam hal ini, metode yang digunakan adalah membedakan objek dengan background. *Local Binary Pattern Histogram* (LBPH) merupakan algoritma kombinasi dari LBP dan *Histogram of Oriented Gradients* (HOG). Pengenalan wajah merupakan pengenalan wajah tingkat tinggi, dalam melakukan pengenalan wajah dapat menggunakan pencocokan dengan LBPH. Gambar wajah yang diambil secara *real time* oleh kamera dibandingkan dan dicocokkan dengan histogram yang diambil dari gambar wajah di *database* (Wibowo, A. W., Karima, A., Wiktasari, A. Y., & Fahriah, S., 2020).



Gambar 2. 12 Konversi biner ke desimal

Dari gambar 2.13 terdapat citra yang telah dikonversi dalam bentuk *grayscale*. Citra tersebut diambil sebagian pikselnya sebesar 3 x 3 piksel yang setiap pikselnya memiliki nilai masing-masing. Perbandingan yang dilakukan menggunakan rumus di bawah ini :

$$LBP = \sum_n^c = S(i_n - i_c)2^n \quad (2.5)$$

(LBP) = Nilai *Local Binary Pattern*

i_c = Nilai piksel pusat

i_n = Nilai tetangga piksel

2.8 OpenCV (*Open Source Computer Vision*)

Library OpenCV (Open Source Computer Vision) merupakan suatu pemrograman aplikasi yang memungkinkan komputer untuk menampilkan objek seperti wajah dan objek lainnya sehingga digunakan untuk pemrosesan citra atau yang disebut dengan *image processing* secara *real-time*. *Library* ini biasa ditujukan untuk *computer vision*, lalu komputer dapat mengambil keputusan, mengambil tindakan, dan mengenali objek yang sarasannya dalam penelitian ini adalah wajah.



Gambar 2. 13 Logo OpenCV

Library OpenCV saat ini bersifat *open source* atau gratis untuk digunakan oleh siapa saja dengan berbagai algoritma *computer vision* (Warnilah, A. I., Jaya-Mulyana, A., Siti-Nuraeni, F., & Aninditya-Widianto, T. 2022).

2.9 Tensorflow

Tensorflow merupakan suatu *framework* yang sudah dikembangkan oleh Google Brain Team pada saat tahun 2015. Saat awal *framework* tensorflow ini dikembangkan digunakan untuk perhitungan numerik. Dengan semakin berkembang nya teknologi, dalam waktu ini *framework* tensorflow biasa dipakai untuk pengembangan aplikasi *Artificial Intelligence* (AI) oleh perusahaan-perusahaan besar. Aplikasi yang telah dikembangkan misalnya seperti pengklasifikasi gambar, penyematan kata, & pengembangan *chatbot* (Wiranda, Purba, Sukmawati, 2020).



Gambar 2. 14 Logo Tensorflow

Framework tensorflow kini telah menyediakan interface yang dapat digunakan untuk mengimplementasikan algoritma *machine learning* dan suatu aplikasi yang digunakan untuk menjalankan algoritma. Algoritma pemodelan yang didukung oleh *framework* ada begitu banyak tensorflow, contohnya adalah *Recurrent Neural Network* (RNN), *Local Binary Pattern Histogram* (LBPH), dan eksekusi paralel (Wiranda , Purba, Sukmawati, 2020).

2.10 Python

Python adalah bahasa yang sangat terkenal di antara pengembang *machine learning*, *Data scientists* maupun *Data Miner*. Python juga bahasa pemrograman yang interpretatif multifungsi. Python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintak, dan juga python merupakan pemrograman yang bersifat *open source* dan *multiplatform*, ada beberapa *feature* yang dimiliki Python. Hal ini membuat Python sangat mudah dipelajari baik untuk pemula maupun untuk yang sudah menguasai bahasa

pemrograman lain. Bahasa ini muncul pertama kali pada tahun 1991, dirancang oleh seorang bernama Guido van Rossum. Sampai saat ini Python masih dikembangkan oleh Python Software Foundation. hampir semua distronya sudah menyertakan Python di dalamnya. Python digunakan sebagai bahasa yang diimplementasi karena kemudahannya. Python lebih banyak diminati karena ia termasuk bahasa pemrograman yang umum sehingga pengembang dapat membuat aplikasi untuk Enterprise lebih mudah (Id, I. D. 2021).



Gambar 2. 15 Logo Pyhton