BAB II
TINJAUAN PUSTAKA

2.1 Penelitian-Penelitian Sebelumnya

Penelitian-Penelitian Sebelumnya adalah istilah yang digunakan untuk menggambarkan perkembangan penelitian terbaru dalam suatu bidang tertentu. Pembahasan tentang penelitian-penelitian sebelumnya penting untuk dilakukan dalam penelitian, karena dapat membantu peneliti untuk memahami apa yang telah dilakukan sebelumnya, serta untuk melihat peluang penelitian baru yang dapat dilakukan.

Penelitian Akhmad Jayadi, Dita Meilinda (2023), berjudul "*Klasifikasi Tingkat Kematangan Buah Pepaya Berdasarkan Warna Kulit Menggunakan Sensor Warna TCS3200*". memiliki Tingkat keberhasilan prediksi tinggi (93% dengan K = 15) Meskipun demikian, penelitian ini memiliki kelemahan, yakni Terbatas pada data yang relatif kecil (100 sampel buah)[3].

Penelitian Lidia Ardhia Wardani, I Gede Pasek Suta Wijaya, Fitri Bimantoro (2022), berjudul "*Klasifikasi Jenis Dan Tingkat Kematangan Buah Pepaya Berdasarkan Fitur Warna, Tekstur Dan Bentuk Menggunakan Support Vector Machine*". Mampu mengklasifikasikan jenis dan tingkat kematangan pepaya secara bersamaan, tetapi memiliki kelemahan yaitu dapat memerlukan waktu komputasi yang lebih lama[5].

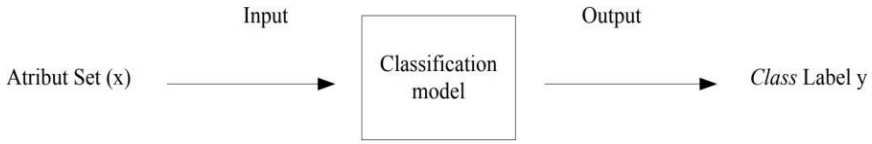
Penelitian Feri Wibowo, Dimara Kusuma Hakim, Sigit Sugiyanto (2018), berjudul "*Pendugaan Kelas Mutu Buah Pepaya Berdasarkan Ciri Tekstur GLCM Menggunakan Algoritma* K-*Nearest Neighbors*". Memiliki kelebihan Tingkat akurasi sebesar 88,88% dengan k = 9, tetapi memiliki kelemahan yaitu keterbatasan pada variasi data yang digunakan dan skala implementasi[6].

Penelitian Alfian Firlansyah, Andi Baso Kaswar, Andi Akram Nur Risal (2021), berjudul "*Klasifikasi Tingkat Kematangan Buah Pepaya Berdasarkan Fitur Warna Menggunakan JST*". memiliki kelebihan Akurasi pengujian sebesar 98,89%

Namun, penelitian ini terbatas pada Dataset yang kecil pada penelitian sebelumnya[7].

2.2 Model Klasifikasi

Klasifikasi adalah tugas pembelajaran sebuah fungsi target f yang memetakan setiap himpunan atribut x ke salah satu label *class* y yang telah didefinisikan sebelumnya. Klasifikasi dapat juga diartikan suatu proses untuk menemukan suatu model atau fungsi yang menggambarkan dan membedakan kelas data atau konsep dengan tujuan dapat menggunakan model untuk memprediksi kelas objek yang label *class*-nya tidak diketahui.



Gambar 2. 1 Model Klasifikasi

(Sumber: Lorena, 2014)

Berdasarkan pada Gambar 2.1 Data *input* untuk klasifikasi adalah isi dari *record*. Setiap *record* dikenal sebagai *instance* atau contoh, yang ditentukan oleh sebuah *tuple* (x, y), dimana x adalah himpunan atribut dan y adalah atribut tertentu, yang dinyatakan sebagai label *class* (juga dikenal sebagai kategori atau atribut target). Pendekatan umum yang digunakan dalam masalah klasifikasi adalah pertama, data *testing* berisi *record* yang mempunyai label *class* yang telah diketahui. Data *training* digunakan untuk membangun model klasifikasi yang kemudian diaplikasikan ke data *testing* yang berisi *record-record* dengan label *class* yang tidak diketahui[8].

2.2.1 Proses Klasifikasi

Klasifikasi data adalah proses mengelompokkan data ke dalam beberapa kelas atau kategori berdasarkan kriteria tertentu, yang membantu analisis, interpretasi, dan pengambilan keputusan berdasarkan data. Ada banyak metode yang dapat digunakan untuk melakukan proses ini. Untuk mengklasifikasikan tingkat kematangan buah pepaya, penulis akan menggunakan teknik KNN.

2.2.2 Jenis-Jenis Metode Klasifikasi

Seperti yang dinyatakan sebelumnya, literatur telah mengusulkan beberapa klasifikasi teknik. Terutama proses klasifikasi dibagi menjadi beberapa kategori yang berbeda, yang dinamakan sebagai keputusan berbasis pengklasifikasi beberapa metode klasifikasi tersebut diantaranya[9]:

1. *Fuzzy*

   Logika *fuzzy* adalah logika yang kabur atau mengandung unsur ketidakpastian. Perkembangan logika ini dimulai di Amerika Serikat pada tahun 1960. Saat ini logika *fuzzy* banyak digunakan di negara-negara maju khususnya Jepang. Logika *fuzzy* digunakan untuk mengontrol berbagai perangkat seperti AC dan mesin cuci. Logika ini cenderung lebih praktis digunakan karena sederhana, mudah dipahami, fleksibel, baik dan ekonomis.

2. *Support Vector Machines*

   SVM merupakan teknik pembelajaran mesin yang beroperasi berdasarkan prinsip minimalisasi risiko struktural (SRM), yang bertujuan untuk menemukan *hyperplane* optimal yang memisahkan dua kelas dalam satu ruang *input*. *Support Vector Machines* (SVMs) pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai serangkaian konsep luar biasa yang harmonis di bidang pengenalan pola.Sebagai teknik pengenalan pola, SVM masih tergolong baru. Namun, evaluasi kemampuannya dalam berbagai aplikasi menunjukkan bahwa ini adalah yang tercanggih di bidang pengenalan pola.

3. *Decision Tree*

   Decision tree merupakan metode klasifikasi yang sering dipakai untuk menyelesaikan masalah. *Decision tree* memiliki beberapa node, yaitu *root node*, *internal node*, dan *leaf node*. Prinsip entropi digunakan untuk menentukan atribut yang akan menjadi titik pemisah (*split*) pada pohon. Pada *decision tree*, setiap *node internal* membagi ruang ke dalam dua bagian atau lebih berdasarkan fungsi diskrit dari nilai atribut *input.*

4. *Naïve Bayes*

   Naïve Bayes Classifier merupakan sebuah metode klasifikasi yang berakar pada teorema *Bayes*. Ciri utama dari *Naïve Bayes Classifier* ini adalah asumsi yang sangat kuat (naïf)akan independensi dari masing-masing kondisi / kejadian. Sebelum menjelaskan *Naïve Bayes Classifier* ini, akan dijelaskan terlebih dahulu Teorema Bayes yang menjadi dasar dari metoda tersebut.

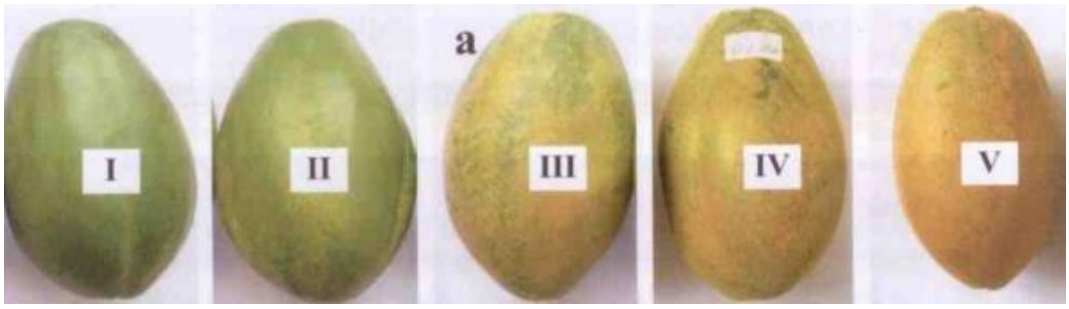
5. K-*Nearest Neighbors*

   Algoritma K-*Nearest Neighbors* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. KNN adalah algoritma *supervised learning*, di mana hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas kategori pada algoritma KNN, di mana kelas yang paling banyak muncul akan menjadi kelas hasil dari klasifikasi ini.

2.3 Buah Pepaya

Pepaya (*Carica papaya L.*) adalah tanaman buah dari suku *Caricaceae* . Tanaman ini berasal dari Amerika tropika dan merupakan hasil dari silang alam antara *Carica peltata Hook. & Arn*. Pepaya sudah dikenal luas dan ditanam di sejumlah daerah tropika dan subtropika di Dunia. Salah satu negara tropis adalah Indonesia dan di setiap daerahnya telah ditanami tanaman pepaya[10].

Terdapat enam stadia kematangan untuk pepaya yaitu munculnya semburat warna kuning pada kulit buah (stadia I), warna kuning 25-49% (stadia II), warna kuning 50- 74% (stadia III), warna kuning diatas 75% (stadia IV), warna kuning penuh 100% (stadia V), dan lewat matang (*over ripe*) lebih jelasnya dapat dilihat pada Gambar 2.2[11].



Gambar 2. 2 Indeks Kematangan Pepaya

(Sumber: Ardiansyah, 2020 mengutip dari Suketi, 2010)

Kandungan gizi pepaya cukup tinggi. Pepaya mengandung berbagai vitamin dan mineral. Kandungan vitamin dalam 100 g pepaya antara lain 474 mg vitamin C, vitamin A 1,094 IU, dan vitamin E 0.73 mg[12]. Kandungan gula utama pada pepaya adalah 48.3% sukrosa, 29.8% glukosa, dan 21.9% fruktosa. Kadar air buah pepaya berkisar antara 81.39-89.21%; 0.29-1.46% protein; 9.65-18.47% karbohidrat; 0.35 0.55 lemak; 14.69-58.78 mg kalsium, dan 6.40-12.80 mg magnesium. Beberapa studi menunjukkan bahwa daun pepaya memiliki efek antisikling dan efektif melawan ulkus gastrik pada tikus, sementara bunga pepaya memiliki aktivitas antibakteri[12].

2.3.1 Klasifikasi tanaman pepaya

Kedudukan tanaman pepaya dalam sistematik (*taksonomi*) tumbuhan diklasifikasikan sebagai berikut :

a. *Kingdom* : *Plantae* (tumbuh-tumbuhan)
b. Divisi : *Spermatophyta* (Tumbuhan berbiji)
c. Sub-Divisi : *Angiosperma* (Biji Tertutup)

d. Kelas : *Dicotyledonae* (Biji berkeping dua)
e. Ordo : *Caricales*
f. Famili : *Caricaceae*
g. Spesies : *Carica papaya L.*

2.4 Pengolahan Citra

Citra digital adalah citra yang dihasilkan dari proses konversi citra analog menjadi citra digital. Konversi ini dilakukan dengan cara mengubah nilai citra analog yang bersifat kontinu menjadi nilai citra digital yang bersifat diskrit. Salah satu alat yang dapat digunakan untuk menghasilkan citra digital adalah kamera digital.

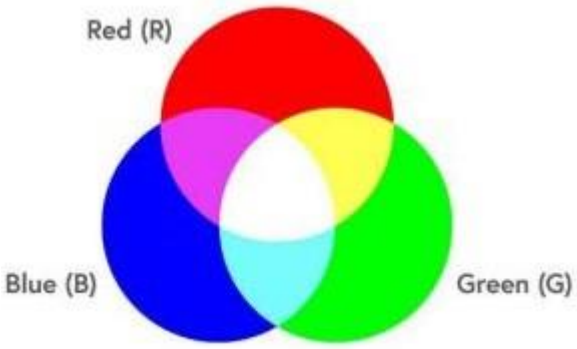
Prinsip dasar pengolahan citra adalah serangkaian teknik yang digunakan untuk meningkatkan kualitas citra, menghilangkan derau, dan mengekstrak informasi dari citra. Tiga prinsip dasar pengolahan citra yang paling umum digunakan adalah peningkatan kontras, penghapusan derau, dan pencarian bentuk objek [13].

Citra digital adalah citra yang diperoleh dari gambar kontinu melalui proses pencuplikan (sampling) secara ruang dan waktu. Pencuplikan adalah proses pengambilan nilai warna pada titik tertentu pada gambar kontinu. Citra digital direpresentasikan sebagai matriks berukuran M × N, di mana setiap elemen matriks menyatakan sebuah piksel. Resolusi citra ditentukan oleh ukuran matriks M × N. Pengolahan citra digital adalah bidang ilmu yang mempelajari cara-cara untuk mengubah citra digital, baik untuk meningkatkan kualitasnya, mengubah penampilannya, atau untuk tujuan analisis [14].

2.4.1 *Red, Green, Blue* (RGB)

RGB adalah warna aditif yang terdiri dari elemen warna merah, hijau, dan biru yang digabungkan dengan warna berbeda untuk membentuk matriks yang mewakili setiap nilai warna.

RGB adalah *color integration* yang terdiri dari tiga integritas warna, yaitu integritas merah (*red*), integritas hijau (*green*), dan integritas biru (*blue*). Setiap integritas mempunyai ukuran set yang tidak sama, ukuran set minimum adalah nol (0) dan ukuran set maksimum adalah 255 (8 bit). Setiap pixel gambar RGB memiliki 16.777.216 pilihan warna (256 x 256 x 256). Representasi warna RGB ditunjukkan pada Gambar 2.3[15].



Gambar 2. 3 Saturasi Warna RGB

(Sumber: Mubarok, 2021)

2.4.2 *Grayscale* (Skala Keabuan)

Citra skala keabuan menawarkan lebih banyak kemungkinan warna dibandingkan citra biner, karena memiliki berbagai nilai antara nilai minimum (hitam) dan nilai maksimum (putih). Banyaknya kemungkinan nilai tersebut bergantung pada jumlah bit yang digunakan. Misalnya, untuk skala keabuan 6 bit, jumlah kemungkinan nilainya adalah 2^6 = 64, dengan nilai maksimum 63. Sedangkan untuk skala keabuan 8 bit, jumlah kemungkinan nilainya adalah 2^8 = 256, dengan nilai maksimum 255. Informasi suatu citra seringkali dapat diwakili oleh histogram ini. Komputasi histogram sangat sederhana dan cepat, serta dapat dilakukan saat memuat citra[16].

2.5 Preprocessing

Preprocessing adalah serangkaian prosedur yang digunakan sebelum gambar *input* diproses ke langkah berikutnya. Tujuan *preprocessing* adalah untuk meningkatkan kualitas gambar dan mempersiapkan gambar untuk tahapan analisis selanjutnya. *Roboflow* adalah salah satu platform yang menyediakan berbagai alat untuk melakukan *preprocessing* citra secara efisien dan efektif. Contoh metode *preprocessing*:

2.5.1 Scaling

Scaling adalah teknik *preprocessing* yang digunakan untuk memastikan bahwa semua gambar dalam dataset memiliki ukuran yang sama. Karena model pembelajaran mesin membutuhkan *input* dengan dimensi yang konsisten untuk berfungsi dengan baik, ini adalah langkah penting[17].

2.5.2 Labeling

Labeling adalah langkah penting dalam *preprocessing* citra, bertujuan untuk memberi identitas atau kelas pada setiap gambar dalam dataset. Proses ini sangat penting untuk tugas klasifikasi karena bertujuan untuk membedakan data berdasarkan label yang nantinya akan digunakan untuk klasifikasi[18].

2.6 Python

Python diciptakan oleh Guido Van Rossum di Belanda pada tahun 1990 dan namanya diambil dari acara televisi kesukaan *Guido Monty Python's Flying Circus. Van Rossum* mengembangkan *python* sebagai hobi, kemudian *python* menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintak sintuitif dan memiliki putaka yang luas[19].

Python adalah bahasa pemrograman interpretatif yang dianggap mudah dipelajari serta berfokus pada keterbacaan kode. *Python* secara umum berbentuk

pemrograman berorientasi objek, pemrograman imperatif dan pemrograman fungsional. Fitur dan kelebihan *python*, yaitu:

1. Memiliki koleksi kepustakaan yang banyak, tersedia modul-modul yang 'siap pakai' untuk berbagai keperluan.
2. Memiliki struktur bahasa yang jelas, sederhana, dan mudah dipelajari.
3. Berorientasi objek.
4. Memiliki sistem pengelolaan memori otomatis.
5. Bersifat modular.

2.6.1 Open CV

OpenCV (*Open Computer Vision*) yaitu sebuah API (*Application Programming Interface*) *library* yang sudah sangat familiar pada pengolahan citra *computer vision*. *Computer vision* merupakan salah satu cabang dari bidang pengolahan citra (*image processing*) yang memungkinkan *computer* dapat melihat seperti manusia.

Open Computer Vision (OpenCV) sendiri merupakan *library open-source* yang tujuannya dikhususkan untuk melakukan pengolahan citra. Maksudnya adalah agar komputer mempunyai kemampuan yang mirip dengan cara pengolahan visual pada manusia. OpenCV telah menyediakan banyak algoritma visi komputer dasar. OpenCV juga menyediakan modul pendeteksian objek[20].

2.6.2 Tensorflow

Tensorflow merupakan salah satu *framework deep learning* dan juga salah satu *library* untuk data *science* yang bersifat *free open-source* yang dikembangkan oleh para peneliti dari tim *Google*. *Tensorflow* dapat digunakan dalam berbagai bidang. Dalam bidang *object detection* terdapat *framework tensorflow object* detection API yang merupakan suatu alat yang dapat digunakan untuk mempermudah proses *constructing*, *training* dan *deployment* pada suatu model *object detection*. *Framework tensorflow object detection* API menyediakan

pretrained object detection model bagi *user*, namun memungkinkan jika *user* ingin menggunakan *pretrained object detection* model yang lain, seperti *Faster* R-CNN, SSD, Retinanet, Resnet50 dan masih banyak lagi[21].

2.6.3 Tkinter

Tkinter (Tk *Interface*) adalah pustaka standar widget GUI (*Graphical User Interface*) untuk membuat antarmuka di *python* menggunakan Tk GUI *Toolkit*. Tkinter adalah pustaka grafis yang memudahkan pembuatan program berbasis grafis. Setiap GUI *Toolkit* menyediakan *widget*, yaitu objek antarmuka pengguna seperti tombol, *scrollbar*, *listbox*, *checkbutton*, *radiobutton*, label teks, dan lainnya. *Widget* mengkapsulasi detail implementasi dan setiap *widget* memiliki perilaku default yang telah didefinisikan, sehingga memudahkan pemrograman GUI[22].

1. Tk: Tkinter adalah pustaka standar *python* yang digunakan untuk membuat antarmuka pengguna grafis (GUI), yang menawarkan berbagai *widget* seperti tombol, label, dan kotak teks untuk membangun aplikasi berbasis GUI. Tkinter juga digunakan untuk membuat jendela utama aplikasi dan membuat berbagai label dan tombol yang digunakan untuk interaksi pengguna.
2. Filedialog: Modul ini memiliki dialog untuk membuka dan menyimpan file. Pengguna dapat memilih gambar untuk dimuat dan diproses oleh aplikasi melalui dialog file.

2.6.4 PIL (*Python Image Library*)

Python Image Library (PIL) berfungsi untuk membuka, menyimpan, dan mengubah gambar yang digunakan dalam proses steganografi. Dengan menggunakan *library* PIL, dapat dilakukan perubahan seperti mengubah ukuran, memotong, memutar, mengubah format, mengubah kualitas, dan banyak lagi[23].

1. Image: Modul ini memungkinkan gambar dibuka, diubah, dan disimpan dalam berbagai format. Pengguna dapat memuat gambar yang dipilih dalam

aplikasi ini dengan PIL dan melakukan berbagai tugas pemrosesan gambar, seperti mengubah ukuran dan mengubah skala keabuan.

2. ImageTk: Modul ini mengubah format objek PIL *Image* sehingga dapat ditampilkan di widget Tkinter. Ini memungkinkan gambar yang dimuat ditampilkan dalam GUI aplikasi.
3. *Image*Ops: Modul ini menawarkan berbagai tugas manipulasi gambar, seperti mengubah ukuran gambar, mengubah skala keabuan, dan membalikkannya.

2.6.5 Numpy (*Numerical Python*)

NumPy (*Numerical Python*) adalah pustaka yang memungkinkan melakukan berbagai tugas manipulasi data dengan *python*. Interaksi antara *array* NumPy dan *python* sangat mirip dengan penggunaan variabel *python* biasa. Untuk tugas yang lebih kompleks, NumPy juga menyediakan fungsi-fungsi untuk aljabar linier, transformasi *fourier*, dan operasi matriks[24].

2.6.6 Pandas

Pandas adalah pustaka yang digunakan untuk manipulasi dan analisis data. Pandas menyediakan struktur data seperti *DataFrame* yang memungkinkan pengolahan dan penyajian data secara efisien. Dalam konteks jurnal ini, Pandas dapat digunakan untuk membaca data dari sumber resmi, memproses data, dan membuat ringkasan statistik[25].

2.6.7 Scikit-Learn

Scikit-Learn adalah pustaka *python* yang dikembangkan untuk memudahkan pengkodean *machine learning* di *python* melalui API yang dibuat oleh berbagai kontributor dari berbagai negara. Pustaka ini sering digunakan di industri dan akademisi. Scikit-Learn dirancang di atas modul NumPy (*Numerical Python*) dan SciPy (*Scientific Python*), yang membuat perhitungannya lebih efisien dan

performanya optimal. Namun, pustaka ini memiliki kelemahan karena tidak dianjurkan untuk digunakan dengan dataset yang sangat besar[26].

2.6.8 Matplotlib

Matplotlib adalah pustaka yang digunakan untuk visualisasi data. Matplotlib menyediakan fungsi dan alat untuk membuat grafik, plot, dan berbagai visualisasi lainnya. Dalam konteks jurnal ini, Matplotlib dapat digunakan untuk membuat histogram, dan visualisasi lain[25].

2.7 Real Time

Real-time object detection (deteksi objek secara *real-time*) menggunakan kamera telah menjadi salah satu aplikasi penting dalam bidang teknologi informasi, khususnya dalam penerapan pada perangkat mobile dan sistem berbasis *deep learning*. Deteksi objek secara *real-time* memungkinkan identifikasi dan klasifikasi objek yang muncul di kamera secara langsung tanpa penundaan yang signifikan, yang sangat penting dalam berbagai aplikasi seperti pengawasan, navigasi kendaraan otonom, dan pemantauan pertanian.

Deteksi objek secara *real-time* mengacu pada kemampuan sistem untuk menganalisis data video yang diperoleh dari kamera dan mendeteksi keberadaan objek dalam waktu yang sangat singkat. Salah satu tantangan utama dalam deteksi *real-time* adalah keseimbangan antara akurasi deteksi dan kecepatan proses[27].

2.8 Deteksi Tepi *Canny*

Deteksi tepi menghasilkan tepi objek gambar untuk menandai bagian yang menjadi detail dan memperbaiki bagian yang kabur karena kesalahan atau efek dari proses pembelian gambar. Dikenal sebagai deteksi tepi yang optimal, algoritma *Canny* memberikan tingkat kesalahan yang rendah. Beberapa metode deteksi tepi, seperti *Canny, Sobel*, dan *Prewitt*, telah dibandingkan dengan hasil metode *Canny* untuk penghitungan deteksi tepi dalam pengolahan kualitas citra. Ini karena output metode *Canny* menunjukkan batas dan tepi yang lebih jelas.

Metode *canny* adalah salah satu algoritma deteksi tepi kontemporer. Marr dan Hildreth, yang melakukan penelitian tentang pemodelan persepsi visual manusia, menemukan deteksi tepi *canny*. Algoritma *canny* dapat memenuhi beberapa kriteria pendeteksi tepian paling ideal, yaitu:

1. Deteksi dengan baik (kriteria deteksi) Kemampuan untuk meletakkan dan menandai semua tepi yang ada sesuai dengan pemilihan parameter—parameter konvolusi yang dilakukan. Selain itu, memberikan fleksibilitas yang sangat tinggi dalam menentukan tingkat deteksi ketebalan tepi yang sesuai dengan keinginan pengguna.
2. Melokalisasi dengan baik (kriteria lokalisasi) *Canny* memungkinkan jarak yang paling sedikit antara tepi yang ditemukan dan tepi yang asli.
3. Respon yang jelas (kriteria respon) Hanya ada satu respons untuk setiap sisi, sehingga mudah dikenali dan tidak membuat gambar menjadi tidak jelas saat diproses[28].

2.9 K-*Nearest Neighbor*

Algoritma k-*Nearest Neighborhood* (k-NN) adalah suatu metode yang menggunakan algoritma *supervised* dimana hasil dari *query instance* yang baru diklasifikasi berdasarkan mayoritas dari label *class* pada k-NN. Tujuan dari algoritma k-NN adalah mengklasifikasi objek baru berdasarkan atribut dan data *training*. Algoritma k-NN bekerja berdasarkan jarak terpendek dari *query instance* ke data *training* untuk menentukan k-NN-nya. Salah satu cara untuk menghitung jarak dekat atau jauhnya tetangga menggunakan metode *euclidean distance*. *Euclidean distance* sering digunakan untuk menghitung jarak. *Euclidean distance* berfungsi menguji ukuran yang bisa digunakan sebagai interpretasi kedekatan jarak antara dua objek, di bawah ini merupakan rumus *euclidean distance*[29]:

$$d(i,j) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \qquad (2\text{-}1)$$

Keterangan:

d = *euclidean*

i = data latih

j = data uji

x = nilai R

y = nilai G

z = nilai B

2.10 Desktop

Desktop merujuk pada antarmuka pengguna grafis yang menampilkan ikon, jendela, dan elemen lain yang memungkinkan pengguna berinteraksi dengan sistem operasi komputer. Biasanya, *desktop* menjadi area kerja utama di mana aplikasi dan dokumen dapat diakses dan dikelola. Peran *desktop* sebagai *platform* di mana perangkat lunak klasifikasi dijalankan. Pengguna dapat mengoperasikan aplikasi klasifikasi, mengatur dataset, dan memvisualisasikan hasil klasifikasi melalui *desktop*.

Optimalisasi *desktop* untuk klasifikasi mendukung proses klasifikasi, *desktop* dapat dioptimalkan dengan cara menyediakan akses cepat ke alat-alat analisis data, memastikan ketersediaan sumber daya komputasi yang memadai, dan memungkinkan integrasi dengan perangkat lunak analitik lainnya[30].

2.11 UML (*Unified Modeling Language*)

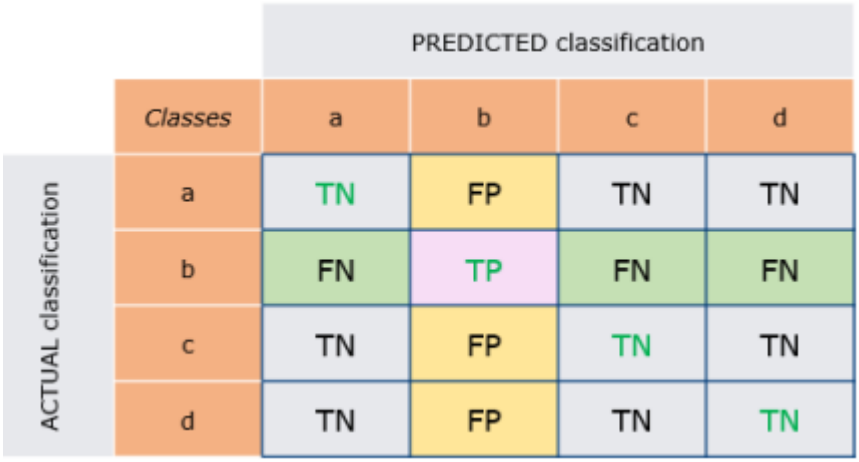
UML atau *Unified Modeling Language* adalah sebuah bahasa standar yang digunakan untuk memodelkan, mendokumentasikan, dan memvisualisasikan sistem perangkat lunak. UML memungkinkan pengembang untuk mengkomunikasikan struktur dan perilaku sistem secara efektif. Kegunaan UML dalam proses pengembangan perangkat lunak untuk membantu analisis dan desain

sistem yang berorientasi objek. UML menyediakan serangkaian diagram yang dapat menggambarkan berbagai aspek sistem, termasuk struktur data, interaksi antar objek, dan proses bisnis.

Diagram UML Terdapat berbagai jenis diagram dalam UML, seperti *use case diagram*, *class diagram*, *activity diagram*, dan *sequence Diagram*. Masing-masing diagram memiliki peranannya dalam memodelkan aspek tertentu dari sistem informasi[31].

2.12 Confusion Matrix

Pada klasifikasi, sumber utama untuk estimasi akurasi adalah matriks konfusi atau *confusion matrix* (matriks klasifikasi atau tabel kontingensi). Gambar 2.4 menunjukkan matriks konfusi untuk masalah klasifikasi dua kelas. Angka-angka sepanjang diagonal dari sudut kiri atas ke sudut kanan bawah mewakili keputusan yang benar, dan angka-angka di luar diagonal mewakili kesalahan. Memperkirakan keakuratan model klasifikasi (atau pengklasifikasi) yang disebabkan oleh algoritma pembelajaran yang diawasi sangat penting karena dua alasan: Pertama, dapat digunakan untuk memperkirakan keakuratan prediksi masa depan, yang dapat menyiratkan bahwa orang harus memiliki tingkat kepercayaan Ini memiliki output dari classifier dalam sistem prediksi. Kedua, dapat digunakan untuk memilih pengklasifikasi dari set yang diberikan (mengidentifikasi model klasifikasi "terbaik" di banyak pelatihan) tabel dapat dilihat pada Gambar 2.4[32].



Gambar 2. 4 *Confusion Matrix*

(Sumber: Grandini, 2020)

- *True Positive* (TP), diartikan memprediksi positif dan benar.
- *True Negative* (TN), diartikan memprediksi negatif dan benar.
- *False Positive* (FP), diartikan memprediksi positif dan salah.
- *False Negative* (FN), diartikan memprediksi negatif dan salah.

Sebagai catatan nilai prediksi adalah *positive* dan *negative*, sedangkan nilai aktual adalah *true* dan *false*. Setelah hasil dari nilai matriks konfusi di dapat, maka performa metriks dapat diukur dengan[32] :

- *Accuracy*, menjelaskan keakuratan model dalam hal ini klasifikasi yang benar.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (2\text{-}2)$$

- *Precision*, menjelaskan akurasi antara data yang diminta dan hasil prediksi yang diberikan oleh model.

$$Precision = \frac{TP}{TP + FP} \quad (2\text{-}3)$$

- *Recall* atau *Sensitivity* (*True Positive Rate*), menjelaskan keberhasilan model dalam mendapatkan informasi.

$$Recall = \frac{TP}{TP + FN} \quad (2\text{-}4)$$

- F-1 *Score*, menjelaskan perbandingan rata-rata bobot *precision* dan *recall* pada saat yang sama.

$$F - 1\ Score = \frac{2\ x\ recall\ x\ precision}{recall + precision} \quad (2\text{-}5)$$