

BAB 2

LANDASAN TEORI

2.1 Dasar Teori

2.1.1 Makanan Tradisional

Berdasarkan Kamus Umum Bahasa Indonesia (Poerwadarminta, n.d.) tradisional memiliki makna sebagai sesuatu yang sifatnya turun temurun dan menurut adat suatu daerah atau kawasan, sedangkan makanan memiliki arti sesuatu yang dimasukkan melalui mulut yang berfungsi memberi kebutuhan nutrisi kepada tubuh sehingga pengertian makanan tradisional secara sederhana berarti sebagai segala sesuatu yang dikonsumsi oleh masyarakat disuatu daerah secara turun temurun guna memenuhi kebutuhan nutrisi bagi tubuhnya.

Pendapat (Ernayanti, 2003) dalam Ensiklopedi Makanan Tradisional di Pulau Jawa dan Pulau Madura memberikan pengertian tentang makanan tradisional memiliki nilai budaya, tradisi, serta kepercayaan yang berasal dari budaya lokal. Hal tersebut berdampak besar pada kebiasaan makan suku bangsa Indonesia, termasuk diantaranya pemilihan bahan mentah, corak dan tradisi makan serta kebiasaan makan dan cara penyajian makanan. Makanan tradisional disuatu daerah bisa menjadi cerminan peradaban dan budaya suatu daerah, akan tepat disuguhkan serta dinikmati oleh masyarakat setempat.

Dari beberapa pengertian tersebut, makanan tradisional Indonesia dapat diartikan sebagai makanan yang diolah dan dibuat oleh masyarakat lokal secara turun temurun dengan menggunakan perpaduan bumbu lokal rempah-rempah khas Nusantara yang memiliki rasa dan aroma spesifik.

2.1.2 Kalori

Kalori adalah satuan ukuran energi. Setiap makanan yang kita makan mengandung sejumlah kalori yang dibutuhkan oleh tubuh untuk melakukan suatu aktivitas. Kalori dalam makanan disediakan oleh karbohidrat, protein, dan lipid. Dari ketiganya, lemak mengandung kalori paling banyak. Setiap gram lemak mengandung 9 kalori, sedangkan setiap gram protein dan karbohidrat mengandung 4 kalori. Kandungan gizi dalam 100 gram nasi putih mengandung 180 kkal, karbohidrat 40,6 gram, lemak 0,1 gram, dan protein 2,1 gram. Nasi putih paling sering dimakan oleh masyarakat Asia Tenggara, termasuk Indonesia. Nasi putih bebas gluten dan kolesterol. Meski mengandung vitamin B1 dan mineral, kandungan gizi beras merah secara keseluruhan masih lebih tinggi dibandingkan beras putih. Batas asupan kalori harian adalah sekitar 1800-2500 kkal dan dapat bervariasi berdasarkan usia, berat badan, dan aktivitas. Untuk makanan dengan energi dinyatakan dalam kilogram (KJ), cukup kalikan nilai ini dalam kilogram dengan 4,2 (1 Joule = 4,2 Kalori).

Tubuh membutuhkan energi (disebut kalori) untuk melakukan aktivitas sehari-hari. Kebutuhan kalori harian setiap individu berbeda. Namun secara umum Kementerian Kesehatan RI menetapkan kebutuhan energi individu adalah 2000 kkal/hari (Kementerian Kesehatan Republik Indonesia, 2018). Faktanya, banyak orang yang tidak memperhatikan kalori pada makanan yang mereka konsumsi setiap hari. Jika kebiasaan ini diteruskan akan berdampak pada kegemukan dan kegemukan akan menimbulkan penyakit lain antara lain penuaan, depresi, gangguan mata, tekanan darah tinggi, masalah gigi, infeksi telinga kronis, diabetes, penyakit jantung, asma dan masih banyak lainnya (Lestari Dwi Asih, 2016).

2.1.3 Ruang Warna

a. Citra RGB

Teknik yang dapat digunakan untuk mengolah atau memanipulasi suatu citra dalam bentuk dua dimensi disebut sebagai image processing. Pengolahan citra juga dapat diartikan sebagai segala jenis aktivitas, baik itu koreksi citra, analisis, atau modifikasi, atau juga mengubah suatu gambar (Rafael C. Gonzalez & Richard E. Woods, 2002). Fitur yang dapat diekstraksi dari suatu objek citra antara lain warna, bentuk, ukuran, dan tekstur. Fitur-fitur ini dapat digunakan sebagai parameter untuk membedakan satu objek dengan objek lainnya. Fungsi-fungsi berikut digunakan dalam pekerjaan ini: Citra berwarna, disebut juga citra RGB, adalah jenis citra yang warna-warnanya berupa komponen R (merah), G (hijau), dan B (biru). Setiap komponen warna menggunakan delapan bit (rentang nilai). 0 sampai 255). Dengan demikian, kemungkinan warna yang dapat ditampilkan adalah $255 \times 255 \times 255$ atau 16.581.375 warna (Abdul & Adhi S, 2013). Berikut adalah contoh image processing untuk melakukan proses ekstraksi fitur warna berdasarkan nilai RGB suatu objek.

1. Membaca citra RGB dan menampilkan kanal merah



Gambar 2. 1 Kanal merah

2. Menampilkan kanal hijau



Gambar 2. 2 Kanal Hijau

- Menampilkan kanal biru



Gambar 2. 3 Kanal Biru

- Menampilkan hasil ekstraksi ciri nilai RGB



Gambar 2. 4 Ciri ekstraksi RGB

- Rentang warna yang sangat luas dalam pencampuran warna dapat ditemukan pada warna merah, hijau, dan biru. RGB juga dapat dikatakan sebagai warna primer (pokok). Intersitas pada suatu warna memiliki nilai tersendiri dengan maksimal 255 (8bit).

	1			2			3			4			5			6		
	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
23	255	0	0	255	100	100	255	150	180	255	200	200	255	230	230	255	255	255
22	255	100	0	255	110	20	255	120	100	255	165	155	255	230	210	255	255	255
21	255	150	0	255	140	20	255	180	40	255	220	120	255	240	190	255	255	255
20	255	200	0	255	210	20	255	230	100	255	240	150	255	255	200	255	255	255
19	255	255	0	255	255	20	255	255	80	255	255	150	255	255	210	255	255	255
18	200	255	0	200	255	20	220	255	80	230	255	150	235	255	180	255	255	255
17	150	255	0	150	255	20	200	255	40	220	255	150	230	255	170	255	255	255
16	100	255	0	110	255	20	150	255	40	200	255	150	230	255	160	255	255	255
15	0	255	0	100	255	100	150	255	150	200	255	200	225	255	225	255	255	255
14	0	255	100	20	255	150	100	255	180	190	255	210	235	255	220	255	255	255
13	0	255	150	20	255	180	100	255	200	160	255	210	200	255	220	255	255	255
12	0	255	200	20	255	200	40	255	220	140	255	230	200	255	255	255	255	255
11	0	255	255	20	255	255	40	255	255	150	255	255	180	255	255	255	255	255
10	0	200	255	20	210	255	40	230	255	60	240	255	170	255	255	255	255	255
9	0	150	255	20	180	255	40	200	255	60	230	255	150	255	255	255	255	255
8	0	100	255	20	150	255	40	180	255	60	200	255	140	255	255	255	255	255
7	0	0	255	20	100	255	40	150	255	60	180	255	160	220	255	255	255	255
6	0	0	200	20	80	255	140	100	255	150	150	255	200	200	255	255	255	255
5	0	0	150	20	50	255	180	100	255	220	140	255	230	180	255	255	255	255
4	0	0	100	20	20	255	220	60	255	240	90	255	255	160	255	255	255	255
3	0	0	50	25	60	255	255	100	240	255	150	250	255	180	255	255	255	255
2	0	0	0	20	20	220	255	40	230	255	140	240	255	170	255	255	255	255
1	0	0	0	20	200	255	40	200	255	100	200	255	160	240	255	255	255	255
0	255	0	0	255	20	120	255	80	140	255	100	160	255	140	180	255	255	255

Gambar 2. 5 Daftar Warna RGB

6. Untuk mencari nilai pada proses ciri ekstraksi dengan mencari nilai rata-rata RGB menggunakan rumus sebagai berikut:

$$\text{Sum Red} = x, y[1,1] + \dots \dots \dots + x, y[h, w] \quad (2.1)$$

$$\text{AVG Red} = \frac{\text{Sum Red}}{\text{Ukuran Frame}}$$

$$\text{Sum Green} = x, y[1,1] + \dots \dots \dots + x, y[h, w] \quad (2.2)$$

$$\text{AVG Green} = \frac{\text{Sum Green}}{\text{Ukuran Frame}}$$

$$\text{Sum Blue} = x, y[1,1] + \dots \dots \dots + x, y[h, w] \quad (2.3)$$

$$\text{AVG Blue} = \frac{\text{Sum Blue}}{\text{Ukuran Frame}}$$

b. Citra Grayscale

Citra *grayscale*, citra ini merupakan citra yang digital yang hanya memiliki nilai kanal pada setiap piksel-nya. Artinya nilai komponen merah, hijau dan biru digunakan untuk menunjukkan tingkat intensitas. warna yang dimiliki oleh citra ini terdiri dari hitam, keabuan, dan putih. Tingkatan keabuan dari citra ini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati warna putih. Untuk dapat mengkonversikan citra warna menjadi citra grayscale maka dapat digunakan persamaan berikut.

$$\text{Gray}(x, y) = 0,2989 x R + 0,5870 x G + 0,1141 x B \quad (2.4)$$

Keterangan :

Gray(x,y) = nilai piksel citra *grayscale*

R = nilai piksel kanal warna merah.

G = nilai piksel kanal warna hijau.

B = nilai piksel kanal warna biru



Gambar 2. 6 Contoh citra *Grayscale*

c. Citra Biner

Dalam citra biner, setiap piksel hanya memiliki 2 kemungkinan nilai, seperti salah dan benar, di mana matriks disimpan sebagai 0 (salah) dan 1 (benar). Citra biner dianggap sebagai citra intensitas bertipe khusus yang berisi hitam dan putih. Selain itu, citra biner dapat dinyatakan sebagai citra ber-indeks yang hanya memiliki 2 warna (Abdul & Adhi S, 2013) Piksel dengan tingkat keabuan 10-derajat di atas ambang batas menerima nilai 1, dan sebaliknya, piksel dengan tingkat keabuan lebih rendah dari ambang batas menerima nilai 0. Gambar biner sering muncul sebagai hasil pemrosesan seperti barcode, morfologi, atau pewarnaan. Fungsi binarisasi sendiri adalah untuk mempermudah proses pengenalan pola, karena pola lebih mudah dikenali pada gambar dengan warna yang lebih sedikit. Mengonversi gambar RGB menjadi gambar biner melalui langkah pertama mengubah gambar RGB menjadi gambar skala abu-abu. Citra grayscale kemudian diubah menjadi citra biner.

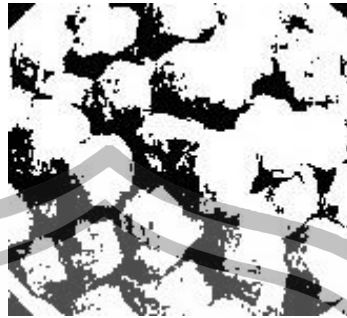
Persamaan yang dapat digunakan untuk mengubah citra grayscale menjadi citra biner adalah sebagai berikut.

$$b(i) = \begin{cases} 0, & 0 \leq i \leq 127 \\ 1, & 128 \leq i \leq 255 \end{cases} \quad (2.5)$$

Keterangan :

$b(i)$ = nilai piksel citra biner

i = nilai piksel *grayscale*



Gambar 2. 7 Contoh citra biner

2.1.4 Edge Detection

a. Metode Sobel

Metode Sobel merupakan pengembangan dari metode Robert dengan menggunakan filter HPF yang diberi buffer nol. Metode Sobel menggunakan prinsip fungsi Laplacian dan Gaussian, yang dikenal sebagai fungsi, untuk membangkitkan HPF. Kelebihan dari metode Sobel adalah kemampuannya dalam mereduksi noise sebelum melakukan perhitungan deteksi tepi. Operator Sobel menggunakan gradien dari operator kernel 3 x 3:

1	2	1
0	0	0
-1	-2	-1

(a)

1	0	-1
2	0	2
1	0	-1

(b)

Gambar 2. 8 Kernel Sobel

$$G_x = [f(i-1, j-1) + 2f(i-1, j) + f(i-1, j+1)] - [f(i+1, j-1) + 2f(i+1, j) + f(i+1, j+1)]$$

$$G_y = [f(i-1, j-1) + 2f(i, j-1) + f(i+1, j-1)] - [f(i-1, j+1) + 2f(i, j+1) + f(i+1, j+1)]$$

$$G[f(x, y)] = \sqrt{G_x^2 + G_y^2} \quad (2.6)$$

Algoritma metode Sobel dalam mendeteksi tepi citra digital adalah sebagai berikut:

- 1) Citra masukan berupa grayscale.
- 2) Konvolusikan citra grayscale dengan kernel Sobel arah horizontal dan vertikal.
- 3) Hitung besar gradient dengan persamaan (2.6).
- 4) Citra keluaran merupakan hasil dari besar gradient (G).

b. *Canny Edge*

Canny edge detector adalah salah satu metode deteksi tepi yang ditemukan oleh John F. Canny, yang digunakan untuk mendeteksi berbagai macam tepian pada gambar. Langkah-langkah deteksi tepi Canny dapat dijelaskan sebagai berikut (Jurusan et al., 2017).

1. Penerapan filter Gaussian

Langkah ini dilakukan untuk menghilangkan noise yang terdapat pada citra. Untuk menghitung nilai setiap elemen dalam gaussian filter menggunakan rumus pada Persamaan

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.7)$$

Keterangan:

$G(x,y)$: nilai elemen gaussian pada posisi (x,y)

π : konstanta dengan nilai 3.14

e : konstanta dengan nilai 2.71828182846

σ : standard deviation

2. Menemukan intensitas gradien citra

Algoritme Canny menggunakan filter untuk mendeteksi tepian horizontal, vertikal, dan diagonal pada sebuah citra. Pada operator deteksi tepi, nilai kembalian merupakan derivatif pertama dalam bentuk arah horizontal (G_x) dan

vertikal (Gy). Pada penelitian ini, digunakan operator Sobel untuk mendapatkan derivatif pertama.

3. Penekanan *non-maximum* digunakan untuk menipiskan garis tepian yang sudah didapatkan dari ketiga langkah sebelumnya. Penekanan ini dilakukan dengan mencari nilai gradien piksel tersebut lalu membandingkannya dengan piksel tetangganya yang termasuk dalam besaran sudut gradien dari piksel tersebut.

- a. Horizontal, bila nilai gradien terletak diantara $22,5^\circ$ hingga $-22,5^\circ$ atau gradien terletak antara $157,5^\circ$ dan $-157,5^\circ$.
- b. Vertikal, bila nilai gradien terletak diantara $-112,5^\circ$ hingga $-67,5^\circ$, atau gradien terletak antara $67,5^\circ$ dan $112,5^\circ$.
- c. $+45$, bila nilai gradien terletak diantara $-67,5^\circ$ hingga $-22,5^\circ$, atau gradien terletak antara $112,5^\circ$ dan $157,5^\circ$.
- d. -45 , bila nilai gradien terletak diantara $-157,5^\circ$ hingga $-112,5^\circ$, atau gradien terletak antara $22,5^\circ$ dan $67,5^\circ$.

Untuk mengubah keluaran dari persamaan (2.6) kedalam satuan derajat, maka digunakan persamaan (2.8).

$$G = \text{atan}\left(\frac{G_y}{G_x}\right) * \frac{180}{\pi} \quad (2.8)$$

4. Menerapkan *thresholding* ganda

Threshold yang digunakan pada algoritma ini digunakan untuk menentukan apakah sebuah edge diklasifikasikan sebagai weak edge atau strong edge. Jika nilai gradien piksel lebih besar dari batas atas, piksel tersebut diklasifikasikan sebagai tepi yang kuat. Jika nilai gradien piksel berada di antara batas atas dan bawah, piksel tersebut adalah tepi yang lemah.



5. Melacak tepian dengan *hysteresis*


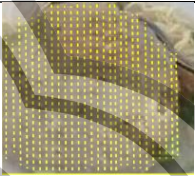


Proses ini dilakukan untuk menjaga tepian agar kontinu. Tepian lemah yang terdeteksi dan bertetangga secara langsung dengan sebuah tepian kuat akan ditandai sebagai tepian kuat.

2.1.5 SMSD (Simple Morphological Shape Descriptors)

SMSD merupakan metode ekstraksi fitur bentuk dari citra. Ekstraksi fitur bentuk yang dihasilkan dari metode SMSD adalah area, length, width, aspect ratio, dan rectangularity N . Area merupakan jumlah piksel yang ada pada objek. Length dan Width merupakan panjang dan lebar dari objek pada citra. Aspect ratio merupakan rasio dari panjang dan lebar yang ada pada objek. Dan rectangularity N merepresentasikan luas minimum persegi panjang yang diisi oleh objek (Wäldchen & Mäder, 2018) Berikut dijelaskan persamaan serta cara kerja pengolahan citra *Simple Morphological Shape Descriptors*.

Tabel 2. 1 Deskripsi Metode SMSD

Deskripsi	Penjelasan	Piktogram
Diameter D	Jarak terjauh antara setiap dua titik pada pinggir objek	 <p>Gambar 2. 9 Ilustrasi Diameter</p>
<i>Major axis length L</i>	Sambungan ruas garis dasar dan ujung objek	 <p>Gambar 2. 10 Ilustrasi Major axis L</p>

<i>Minor axis length W</i>	Lebar maksimum tegak lurus terhadap sumbu utama objek	 Gambar 2. 11 Ilustrasi Minor axis W
<i>Area A</i>	Jumlah piksel dalam wilayah objek	 Gambar 2. 12 Ilustrasi pengambilan area
<i>Perimeter P</i>	Penjumlahan dari jarak antara masing-masing pasangan piksel yang berdampingan di sekitar perbatasan objek	 Gambar 2. 13 Ilustrasi Pengambilan Perimeter
<i>Centroid</i>	Mewakili koordinat geometris pada tengah bagian dari bentuk objek	 Gambar 2. 14 Ilustrasi Pengambilan Centroid

- a. *Aspect ratio* (AR) merupakan rasio perbandingan antara major axis length dengan minor axis length. Fitur ini digunakan untuk memperkirakan bentuk objek makanan. Jika bernilai lebih dari 1 maka objek tersebut melebar. Jika kurang dari 1 maka bentuk

objek tersebut memanjang. Persamaan untuk menghitung aspect ratio dapat dilihat pada Persamaan

$$AR = \frac{L}{W} \quad (2.9)$$

- b. *Roundness* (R) mengilustrasikan perbedaan antara objek dan lingkaran. Fitur ini digunakan untuk mengukur seberapa bundar bentuk objek makanan tersebut. Untuk menghitung roundness dapat dilihat pada Persamaan

$$R = \frac{4\pi A}{P^2} \quad (2.10)$$

- c. *Rectangularity* (N) mengilustrasikan kemiripan antara objek dengan empat persegi panjang. Persamaan untuk menghitung rectangularity dapat dilihat pada Persamaan

$$N = \frac{A}{LW} \quad (2.11)$$

- d. *Compactness* (C) merupakan rasio antara perimeter dengan area objek, fitur ini memberikan informasi tentang kompleksitas umum dan faktor bentuk. Persamaan untuk menghitung compactness dapat dilihat pada Persamaan.

$$C = P \frac{P}{\sqrt{A}} \quad (2.12)$$

- e. *Narrow Factor* (NF) merupakan rasio antara diameter dengan major axis length. Persamaan untuk menghitung narrow factor dapat dilihat pada Persamaan

$$NF = \frac{D}{L} \quad (2.13)$$

- f. Rasio perimeter dengan diameter (PD) merupakan ukuran perbandingan perimeter dengan diameter. Persamaan untuk menghitung rasio perimeter dengan diameter dapat dilihat pada Persamaan.

$$PD = \frac{P}{D} \quad (2.14)$$

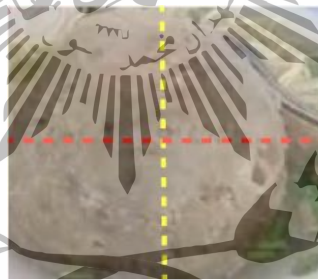
- g. Rasio *perimeter* dengan *major axis length* (PL) merupakan ukuran perbandingan *perimeter* dengan *major axis length*. Persamaan untuk menghitung Rasio *Perimeter* dengan *Major Axis Length* dapat dilihat pada Persamaan.

$$PL = \frac{P}{L} \quad (2.15)$$

- h. Rasio *Perimeter* dengan *Major axis length* dan *Minor axis length* (PLW) merupakan rasio *perimeter* dengan jumlah *major axis length* dan *minor axis length*. Persamaan untuk menghitung Rasio *Perimeter* dengan *Major axis length* dan *Minor axis length* dapat dilihat pada Persamaan.

$$PLW = \frac{P}{(L+W)} \quad (2.16)$$

Pada skripsi ini mengambil beberapa fitur SMSD yang akan dijadikan sebagai ciri ekstraksi



Gambar 2.15 Ilustrasi menghitung *aspect ratio*

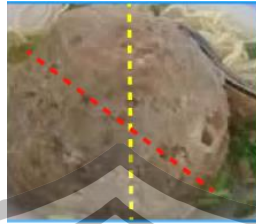
Ciri ekstraksi *aspect ratio* di hitung menggunakan persamaan (2.9)

$$\text{aspect ratio} = \frac{L}{W}$$

Keterangan:

W = Fitur lebar.

L = Fitur panjang



Gambar 2. 16 Ilustrasi menghitung *rectangularity*

Ciri ekstraksi *rectangularity* di hitung menggunakan persamaan

(2.11)

$$\text{rectangularity} = \frac{A}{LW}$$

Keterangan:

A = Area

W = Fitur lebar.

L = Fitur panjang



Gambar 2. 17 Ilustrasi menghitung nilai *Compactness*

Ciri ekstraksi *Compactness* di hitung menggunakan persamaan

(2.12)

$$C = P \frac{P}{\sqrt{A}}$$

Keterangan:

A = Area

P = Perimeter



Gambar 2. 18 Ilustrasi menghitung nilai *Roundness*

Ciri ekstraksi *Roundness* di hitung menggunakan persamaan (2.10)

$$R = \frac{4\pi A}{p^2}$$

Keterangan:

A = Area

P = Perimeter

$\pi = 3,14$

2.1.6 Klasifikasi

Klasifikasi adalah proses menemukan pola yang menjelaskan atau membedakan konsep atau kelas informasi, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang kelasnya tidak diketahui (Pang-Ning Tan et al., 2020). Di dalam klasifikasi diberikan sejumlah record yang dinamakan training set, yang terdiri dari beberapa atribut, atribut dapat berupa kontinyu ataupun kategoris, salah satu atribut menunjukkan kelas untuk record. Tujuan dari klasifikasi adalah untuk:

1. Setelah ditemukan pola dalam training set yang memisahkan record ke dalam kelas atau kategori yang sesuai, pola tersebut kemudian digunakan.
2. Mengklasifikasikan record yang kelasnya tidak diketahui sebelumnya dalam set pengujian.
3. Mengambil keputusan dengan memprediksi kasus berdasarkan hasil klasifikasi yang diperoleh. Klasifikasi dibagi menjadi dua tahap, yaitu pelatihan dan prediksi. Pada fase pelatihan, model

area masalah dibuat dari setiap dataset/kasus yang ada. Spesifikasi model didasarkan pada analisis sekumpulan data pelatihan, yaitu data dengan label kelas yang diketahui. Pada fase klasifikasi, prediksi kelas dibuat dari kasus/data uji baru yang dibuat.

a. Metode *K-Nearest Neighbor* (KNN)

Algoritma *k-nearest neighbor* (KNN) adalah metode untuk mengklasifikasikan objek berdasarkan data training yang paling dekat dengan objek tersebut. Prinsip kerja dari *K-Nearest Neighbor* (KNN) adalah mencari jarak terdekat antara data yang akan dievaluasi dengan k tetangga (*neighbor*) terdekatnya dalam data pelatihan (Sreemathy & Balamurugan, 2022).

Pada tahap pembelajaran, algoritma ini hanya menyimpan vektor fitur dan mengklasifikasikan data pembelajaran. Pada tahap klasifikasi, fitur yang sama dihitung untuk data uji. Mari hitung jarak vektor baru ini ke semua vektor data pelatihan dan ambil k bit terdekat. Nilai k terbaik untuk algoritma ini bergantung pada data. Secara umum, nilai k yang tinggi mengurangi efek noise pada klasifikasi, tetapi mengaburkan batas antara setiap klasifikasi. Nilai k yang baik dapat dipilih dengan mengoptimalkan parameter, misalnya dengan validasi silang. Kasus khusus di mana klasifikasi diprediksi berdasarkan data pelatihan terdekat (yaitu $k=1$) disebut algoritma *k-nearest-neighbor*. Kelebihan dan kekurangan metode JST adalah sebagai berikut:

1. Kelebihan KNN memiliki kelebihan yaitu tangguh terhadap training data yang noise dan efektif apabila data latih nya besar.

2. Kelemahan

- a. KNN perlu menentukan nilai dari parameter k (jumlah dari tetangga terdekat).
- b. Pembelajaran berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil yang terbaik.
- c. Upaya komputasi cukup tinggi karena perlu menghitung jarak dari setiap sampel uji ke seluruh sampel pelatihan.

Tujuan dari algoritma KNN adalah untuk mengklasifikasikan objek baru berdasarkan atribut dan pola pelatihan, dimana hasil dari pola pengujian yang direklasifikasi didasarkan pada k kelas tetangga terdekat yang paling banyak. Dalam proses klasifikasi, algoritma ini tidak menggunakan model untuk pencocokan dan hanya mengandalkan memori. Algoritma JST menggunakan klasifikasi tetangga sebagai nilai prediksi dari sampel uji baru. Menurut (Sreemathy & Balamurugan, 2022), peringkat untuk k tetangga terdekat berdasarkan nilai kesamaan dihitung menggunakan jarak Euclidean (*Euclidean Distance*) yang didefinisikan sebagai berikut:

$$D(X,Y) = \sqrt{\sum_{i=1}^p (x_i - y_i)^2} \quad (2.17)$$

Keterangan:

D (X,Y):Jarak *Euclidean*(*Euclidean Distance*)

x_i :sampel data

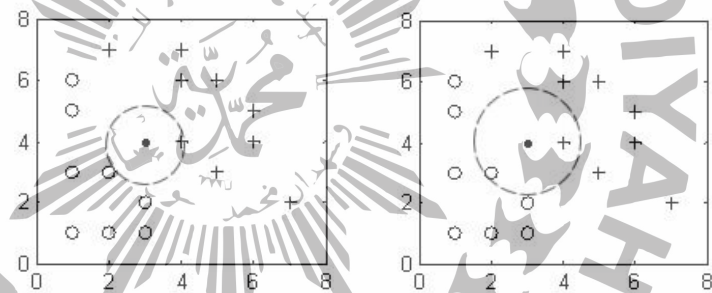
y_i :data uji

P:dimensi data

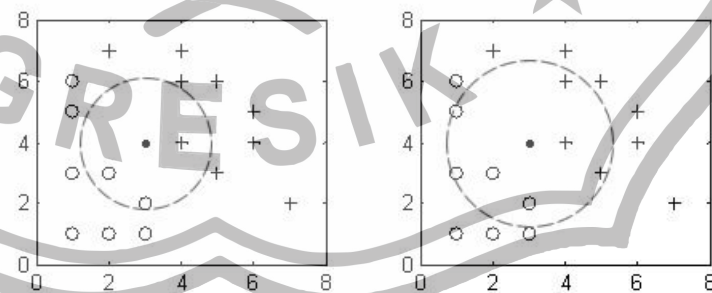
I:variabel data

Adapun langkah-langkah klasifikasi algoritma KNN adalah sebagai berikut:

1. Tentukan parameter nilai k = banyaknya jumlah tetangga terdekat
2. Hitung jarak antara data baru dengan semua data training
3. Urutkan jarak dan tetapkan tetangga terdekat berdasarkan jarak minimum ke- k
4. Periksa kelas dari tetangga terdekat
5. Gunakan mayoritas sederhana dari kelas tetangga terdekat sebagai nilai prediksi data baru



1 tetangga terdekat (1-NN) 2 tetangga terdekat (2-NN)



3 tetangga terdekat (3-NN) 7 tetangga terdekat (7-NN)

Gambar 2. 19 ilustrasi algoritma KNN

Dari ilustrasi pada gambar dapat diketahui jika semakin dekat jarak data training terhadap data testing, maka data training tersebut merupakan data yang paling mirip dengan data testing.

2.1.7 Akurasi

Dalam mengevaluasi performansi sebuah model yang dibangun oleh algoritma klasifikasi dapat dilakukan dengan menghitung jumlah dari test record yang di prediksi secara benar (akurasi) atau salah (error rate) oleh model tersebut. Akurasi didefinisikan sebagai berikut.

$$\text{Akurasi} = \frac{\text{Jumlah prediksi benar}}{\text{Jumlah total prediksi}} \times 100 \quad (2.18)$$

2.1.8 Sistem Operasi *Android*

Sistem operasi *Android* telah digunakan oleh milyaran manusia di dunia. Sistem operasi yang mulanya dibuat untuk smartphone HP/Tablet ini dikembangkan oleh Google dan banyak perusahaan besar di dunia dengan cara gotong royong atau Open Source. App Inventor 2 (AI2) adalah aplikasi web opensource yang dikembangkan oleh Google.

2.1.9 *Android Studio*

Android Studio adalah *Integrated Development Environment (IDE)* atau software yang bisa digunakan untuk mengembangkan aplikasi android. *Software* ini diperkenalkan pertama kali pada tahun 2013 diacara Google I/OConference. *Android Studio* merupakan *software* resmi yang didukung penuh oleh Google sebagai perusahaan induk Sistem Operasi *Android*. IDE ini dikembangkan oleh JetBrains dan dirilis pertama kali ke publik pada tahun 2014.

2.2 Kajian Pustaka

Pada bab 2 ini penulis akan membahas mengenai teori-teori yang akan digunakan dalam klasifikasi jumlah kalori makanan Tradisional. Pada bagian ini akan menjelaskan tentang teori-teori yang digunakan dalam proses klasifikasi.

Penelitian yang telah dilakukan sebelumnya yaitu tentang Klasifikasi Kualitas Wortel Menggunakan Metode *K-Nearest Neighbor* Berbasis Android oleh (Al Azami et al., 2022). Pada penelitian ini bertujuan untuk mengetahui mana wortel yang berkualitas baik dan mana yang kurang berkualitas buruk serta bukan wortel dengan menggunakan ciri ekstraksi yang diambil yaitu warna RGB (*Red, Green, Blue*) dan HSL (*Hue, Saturation, Lightness*). Penelitian ini memberikan hasil bahwa penerapan metode KNN dan untuk klasifikasi kualitas wortel bisa berjalan dengan baik. Dan pada hasil klasifikasi menggunakan sistem klasifikasi ini mendapat presentase sebesar 74,19%.

Penelitian lain yang telah dilakukan ada penelitian tentang Penggunaan *K-Nearest Neighbor (KNN)* untuk Klasifikasi Citra Makanan Menggunakan *K-Nearest Neighbor* dengan Fitur Bentuk Simple Morphological Shape Descriptors dan Fitur Warna Grayscale Histogram oleh (Setiawan et al., 2019). Penelitian ini menunjukkan hasil Klasifikasi Citra Makanan juga dapat mengklasifikasikan citra makanan dengan baik Pada penelitian ini digunakan ekstraksi ciri *Simple Morphological Shape Descriptors (SMSD)* dan menghasilkan fitur *area, length, width, aspect ratio, rectangularity N*. Setelah didapatkan hasil ekstraksi fitur, dilakukan klasifikasi menggunakan metode *K-Nearest Neighbor*. Berdasarkan hasil pengujian jika hanya menggunakan metode *Grayscale Histogram* menghasilkan nilai akurasi sebesar 60%. Jika hanya menggunakan metode SMSD menghasilkan nilai akurasi sebesar 54,8%. Jika menggunakan metode

Grayscale Histogram dan metode SMSD menghasilkan nilai akurasi sebesar 77,8%.

Penelitian dengan judul Penggunaan *K-Nearest Neighbor* (KNN) untuk Mengklasifikasi Citra Belimbing Berdasarkan Fitur Warna (Imantata Muhammad & Falih, 2021). Penelitian ini memberikan hasil fitur warna bisa mengidentifikasi tingkat kematangan buah belimbing berdasarkan citra dengan algoritma *K-Nearest Neighbor* dan dengan menggunakan metode ekstraksi ciri *Hue saturation Value* (HSV). Dengan menggunakan algoritma KNN didapatkan akurasi sebesar 93.33% pada percobaan dengan menggunakan nilai $K=7$.

Penelitian berjudul Klasifikasi Kualitas Jeruk Lokal Berdasarkan Tekstur dan Bentuk Menggunakan Metode *K-Nearest Neighbor* (K-NN) (Enggar Pawening et al., 2020). Tujuan dari penelitian ini adalah untuk mengklasifikasikan jeruk lokal berdasarkan bentuk dan teksturnya Metode yang digunakan adalah metode GLCM (*Gray Level Co-Occurrence Matriks*) yang juga meliputi metode energi, korelasi, kontras, homogenitas dan pemisahan fitur struktural termasuk, seperti metode geometris. Penghapusan fitur dengan fitur eksentrisitas dan matriks. Sementara itu, menentukan kedekatan citra uji dan citra latih berdasarkan fitur tekstur dan bentuk yang diperoleh dengan metode k-nearest. Hasil dari penelitian ini Metode knn berhasil mengidentifikasi jeruk dengan baik dan buruk berdasarkan bentuk dan teksturnya. Akurasi tertinggi yang didapat dalam penelitian ini mendapatkan nilai 93,33% pada uji $k=1$, sedangkan untuk yang terendah 86,20% pada uji $k=7$ dan $k=8$.

Penelitian dengan judul Implementasi *K-Nearest Neighbor* Untuk Klasifikasi Bunga Dengan Ekstraksi Fitur Warna RGB (Farokhah & Korespondensi, n.d.). Tujuan Penelitian ini adalah mengidentifikasi jenis bunga Dalam evaluasi klasifikasi KNN yang dilakukan dengan ekstraksi fitur warna RGB terhadap empat jenis label bunga yaitu bunga coltsfoot, bunga daisy, bunga dandelion dan bunga matahari. Hasil dari penelitian ini menunjukkan bahwa kolaborasi metode klasifikasi KNN dengan ekstraksi

fitur warna RGB memiliki kelemahan terhadap percobaan pertama dengan akurasi 50-60% pada $K=5$. Percobaan kedua memiliki akurasi sekitar 90-100% pada $K=5$. Peningkatan akurasi, precision dan recall terjadi ketika menaikkan jumlah K yaitu dari $K=1$ menjadi $K=3$ dan $K=5$.

Penelitian dengan judul Ekstraksi Citra menggunakan Metode GLCM dan KNN untuk Identifikasi Jenis Anggrek (*Orchidaceae*) (Putra Pamungkas et al., 2019). Tujuan penelitian ini adalah untuk menghasilkan aplikasi yang dapat mengidentifikasi jenis bunga anggrek. Data yang digunakan sebagai data training pada aplikasi yang dibangun berjumlah 75 data dengan perincian 25 data bunga jenis *phalaenopsis*, 25 data jenis *dendrobium*, 25 data bukan anggrek. Data Testing berjumlah 15 dengan perincian 5 data bunga jenis *phalaenopsis*, 5 data jenis *dendrobium*, 5 data bukan anggrek. Hasil dari pengujian dan analisa dari penelitian yang telah dilakukan dapat disimpulkan tingkat keberhasilan identifikasi *Orchidaceae* atau bunga anggrek mencapai 80% dengan rata-rata 77%. Nilai K berpengaruh pada tingkat keberhasilan identifikasi, semakin besar nilai K akurasi semakin kecil.

Penelitian dengan judul Klasifikasi Jenis Bawang Menggunakan Metode *K-Nearest Neighbor* Berdasarkan Ekstraksi Fitur Bentuk dan Tekstur (Angelina Widians et al., 2019). Tujuan penelitian ini adalah untuk mengidentifikasi jenis bawang Berdasarkan hasil penelitian klasifikasi jenis bawang menggunakan metode *k-nearest neighbor* berdasarkan ekstraksi fitur bentuk dan tekstur menghasilkan kesimpulan. Rata-rata akurasi penggunaan nilai k pada 5 kali pengujian untuk $k3$ 73% dan $k5$ mendapatkan 81.80% sedangkan yang terbaik yaitu pada $k7$ dengan akurasi 83.13%.

Penelitian yang menggunakan Citra biner dengan judul Pengaruh Nilai K Pada Metode *K-Nearest Neighbor* (KNN) Terhadap Tingkat Akurasi Identifikasi Kerusakan Jalan (Angreni et al., 2019). Tujuan penelitian adalah berapa besar pengaruh nilai k dari metode KNN terhadap tingkat akurasi jenis kerusakan retak dan retak kulit buaya dengan menggunakan proses pengolahan citra biner Berdasarkan uji coba yang dilakukan terhadap 100

citra dengan kerusakan retak dan retak kulit buaya masing-masing 50 citra, didapat nilai akurasi yang berbeda. Untuk nilai $k=1$, akurasi untuk retak = 98% dan retak kulit buaya = 84%. Untuk nilai $k=8$, akurasi untuk retak = 96% dan retak kulit buaya = 8%. Untuk nilai $k=15$, akurasi untuk retak = 0% dan retak kulit buaya = 98%.

Penelitian yang menggunakan Deteksi tepi Canny dengan judul Analisis Pengenalan Pola Daun Berdasarkan Fitur Canny Edge Detection dan Fitur GLCM Menggunakan Metode Klasifikasi *k-Nearest Neighbor* (kNN) (Azizah Arif Paturrehman, 2021). Berdasarkan hasil pengujian yang telah dilakukan, pengembangan pengenalan pola daun sudah berjalan dengan baik dengan menggunakan fitur Canny dan GLCM, Informasi citra daun yang digunakan dalam riset ini berupa dataset sebanyak 350 citra yang terdiri dari 7 spesies dengan 50 citra daun tiap spesies. Pengujian ini dilakukan dengan metode ekstraksi fitur deteksi tepi Canny dengan 4 fitur yaitu area, perimeter, eccentricity dan matric. Pada pengujian ini menggunakan tiga variasi nilai k yaitu k_3 , k_5 dan k_7 mendapatkan akurasi tertinggi 98% pada k_3 dengan perbandingan 70:30 data training dan testing.

Penelitian yang menggunakan Deteksi tepi Canny dengan judul Implementasi Metode Canny Untuk Deteksi Tepi Mutu Daun Tembakau oleh (Yudiyanto & Ahmad Dahlan Soepomo, 2014). Berdasarkan hasil pengujian yang telah dilakukan, maka diperoleh hasil dari deteksi tepi menggunakan metode Canny menghasilkan titik tepian yang bersambung cukup jelas sehingga mempermudah dalam uji mutu daun tembakau. Dari hasil pengujian program dengan menggunakan 30 sampel citra daun tembakau didapat akurasi kebenarannya sebesar 90% sehingga aplikasi ini layak digunakan.

Penelitian yang menggunakan Metode KNN dan menggunakan ciri ekstraksi Area dan Perimeter berikutnya dengan judul. Klasifikasi Kualitas Fisik Kopi Arabika Menggunakan Pengolahan Citra Dengan Metode K-Nearest Neighbor(K-NN) oleh (Mardisa et al., 2022) yang menggunakan ciri ekstraksi ukuran dan bentuk (*area, perimeter*), ekstraksi ciri tekstur (kontras,

korelasi, energi, homogeneity) dan ekstraksi ciri warna (R, G, B, L, a, b) mendapatkan hasil penelitian menunjukkan bahwa dengan menggunakan nilai $K=5$ nilai rata-rata akurasi yaitu 78,625% sedangkan menggunakan $K=3$ dengan nilai rata-rata akurasi yaitu 58,000%.

