

**PENGENALAN CITRA JENIS MAKANAN MENGGUNAKAN  
EKSTRAKSI FITUR *COLOR CHANNEL* DAN *GRAY LEVEL CO-  
OCCURENCE MATRIX***

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Ofi Eka Novyanti

NIM: 155150207111101



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2019

## PRAKATA

Dengan menyebut nama Allah SWT Yang Maha Pengasih Lagi Maha Penyayang. Syukur Alhamdulillah berkat pertolongan, kasih sayang, dan izin Yang Kuasa, penulis dapat menyelesaikan tugas akhir skripsi dengan judul “Pengenalan Citra Jenis Makanan Menggunakan Ekstraksi Fitur *Color Channel* dan *Gray Level Co-occurrence*” tepat waktu. Tak lupa pula shalawat serta salam penulis junjatkan kepada Nabi Muhammad SAW, keluarga beserta sahabat. Melalui kesempatan ini penulis sampaikan rasa hormat dan terimakasih yang sebesar-besarnya kepada pihak yang telah memberikan dukungan serta bantuan dalam penyelesaian skripsi ini. Pihak-pihak tersebut antara lain:

1. Ibu Yuita Arun Sari, S.Kom., M.Kom dan Bapak Muhammad Tanzil Furqon, S.Kom., M.CompSc selaku dosen pembimbing I dan dosen pembimbing II yang telah banyak memberikan bimbingan, ilmu, masukan dalam penyusunan skripsi ini.
2. Keluarga dari penulis terutama kedua orang tua dan adik yang penulis yang telah memberikan doa, motivasi, kasih sayang, serta dukungan materil.
3. Sisco Jupiyandi yang telah sepenuh hati menemani, mendukung, serta memberikan semangat dan dukungan kepada penulis selama masa perkuliahan.
4. Vriza Wahyu Saputra, M. Rizky Setiawan, Ahmad Fauzi Ahsani, dan Sarah Najla Adha, yang telah begitu sabar mengajari dan membantu penulis pada pengambilan dataset hingga pengoreksian penulisan.
5. Sahabat-sahabat Multichat yang terkasih Hanoum Eva Hayati, Yugi Trilia Septiana, Muhammad Enowato, M. Thareq P. B., Septian, M. Rizky Setiawan, Abd. Jahiddudin, Ruri, Bagas, Ilham, Oni, Firda, dan Marsha terima kasih untuk canda tawa, tangis, perhatian, serta drama yang telah mengisi hari-hari sejak semester 5 perkuliahan.
6. Teman-teman Induksi Riset 2015 Devinta S. Atmaja, Karunia A., Gregorius Ivan Sebastian, dan banyak teman yang lain yang tidak bisa saya sebutkan satu-satu. Terima kasih atas waktu dan perhatiannya selama ini sehingga penulis selalu terpacu untuk menyelesaikan skripsi dengan cepat.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran senantiasa penulis harapkan dari berbagai pihak untuk kesempurnaan skripsi ini.

Malang, 6 Maret 2019

Penulis

ofinovyanti@student.ub.ac.id

## ABSTRAK

**Ofi Eka Novyanti, Pengenalan Citra Jenis Makanan Menggunakan Ekstraksi Fitur *Color Channel* dan *Gray Level Co-Occurrence Matrix***

**Pembimbing: Yuita Arum Sari, S.Kom., M.Kom dan Muhammad Tanzil Furqon S.Kom, M.CompSc**

Secara manual, manusia dapat dengan mudah mengenali dan membedakan jenis-jenis makanan. Akan tetapi sistem penglihatan manusia tentu memiliki keterbatasan sehingga identifikasi jenis makanan yang dihasilkan berbeda-beda dan tidak konsisten. Penelitian menunjukkan bahwa sistem penglihatan manusia dipengaruhi oleh beberapa faktor seperti usia dan penyakit mata serta faktor kondisi psikis manusia. Seiring berjalannya teknologi, banyak penelitian bermunculan terkait sistem pengenalan jenis makanan menggunakan pengolahan citra. Penelitian ini bertujuan untuk mengembangkan suatu sistem yang dapat mengenali jenis makanan berbasis citra berdasarkan warna dan teksturnya menggunakan metode *K-Nearest Neighbor* (K-NN). Prosesnya diawali dengan mensegmentasi pada citra untuk mendapatkan fokus objek citra. Objek citra tersegmentasi tersebut kemudian digunakan untuk mendapatkan nilai ekstraksi fitur warna menggunakan *color channel* RGB sebanyak 9 sub fitur dan fitur tekstur dengan *Gray Level Co-Occurrence Matrix* (GLCM) sebanyak 20 sub fitur pada sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$ . Hasil proses ekstraksi fitur selanjutnya digunakan dalam proses identifikasi citra menggunakan metode K-NN. Pengujian dilakukan melalui 3 tahap yaitu pengujian nilai k, pengujian ekstraksi fitur, dan pengujian perhitungan jarak tetangga terdekat pada 450 dataset untuk 2 jenis kategori data yaitu data seimbang dan data tidak seimbang. Hasil dari pengujian ini didapatkan akurasi terbaik untuk nilai  $k=3$  sebesar 90,58% pada data seimbang.

**Kata Kunci:** Pengenalan Citra Makanan, *Color Channel*, *Gray Level Co-Occurrence Matrix*, *klasifikasi*

## ABSTRACT

**Ofi Eka Novyanti, Pengenalan Citra Jenis Makanan Menggunakan Ekstraksi Fitur *Color Channel* dan *Gray Level Co-Occurence Matrix***

**Supervisors: Yuita Arum Sari, S.Kom., M.Kom dan Muhammad Tanzil Furqon S.Kom, M.CompSc**

*Manually, humans can easily detect and differentiate types of food. But man sees certainly has contrains that meant identification a kind of food produced different and inconsistant. Some researches show that human vision system influenced by many factors, such as age and disease and human's physical condition. Through the passing technology, researchers have seen the emergence of food's recognition systems related with image processing. The aim of this study is to develop a system that could recognize many kinds of food based on it's color and texture by using K-Nearest Neighbor (K-NN). The process was proceded by segmentation on images to get the focus object of the image. The segmented image then used to gain the color feature extraction's value, color channels that use RGB as the channel with 9 sub features and the texture value of feature extraction, grey level co-occurence matrix (GLCM) with 20 sub features at angle 0°, 45°, 90°, and 135°. The results of the process of extracted feature then used in the process of image classification using K-NN method. Testing process done through 3 stages of tests that are k value testing, feature extraction testing, distance calculation method testing with 450 data sets of two types of data categories. The best result of testing process is data which use value k =3 that earn as much as 90,58% of accuration with balanced composition data.*

**Keywords:** *Food Image Recognition, Color Channel, Grey Level Co-Occurence Matrix, classification*

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT .....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR KODE PROGRAM .....	xii
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan .....	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	4
<b>BAB 2 LANDASAN KEPUSTAKAAN .....</b>	<b>6</b>
2.1 Kajian Pustaka .....	6
2.2 Pengolahan Citra Digital .....	9
2.2.1 Segmentasi Citra .....	9
2.3 Ruang Warna ( <i>Color Channel</i> ).....	12
2.4 <i>Euclidean Distance</i> .....	13
2.5 <i>Manhattan Distance</i> .....	13
2.6 <i>Gray Level CO-OCCURENCE MATRIX (GLCM)</i> .....	13
2.7 Metode <i>K-Nearest Neighbor (K-NN)</i> .....	14
2.7.1 Konsep <i>K-Nearest Neighbor (K-NN)</i> .....	15
2.8 <i>K-Fold Cross Validation</i> .....	15
2.9 Akurasi .....	16
<b>BAB 3 METODOLOGI .....</b>	<b>17</b>
3.1 Studi Literatur .....	17



3.2 Pengumpulan Data.....	18
3.3 Analisis dan Perancangan .....	18
3.4 Implementasi .....	18
3.5 Pengujian dan Analisis.....	18
3.6 Kesimpulan.....	19
3.7 Spesifikasi <i>Hardware</i> dan/atau <i>Software</i> .....	19
<b>BAB 4 PERANCANGAN.....</b>	<b>20</b>
4.1 Perancangan Algoritme .....	20
4.1.1 <i>Pre-Processing</i> .....	21
4.1.2 Ekstraksi Tekstur GLCM.....	22
4.1.3 Ekstraksi Warna dengan <i>Color Channel</i> .....	24
4.2 Perhitungan Manual .....	28
4.2.1 Ekstraksi Tekstur GLCM.....	28
4.2.2 Ekstraksi Warna ( <i>Color Channel</i> ).....	39
4.2.3 Klasifikasi K-NN.....	41
4.3 Perancangan Skenario Pengujian .....	43
<b>BAB 5 IMPLEMENTASI .....</b>	<b>45</b>
5.1 Lingkungan Implementasi.....	45
5.1.1 Lingkungan <i>Hardware</i> .....	45
5.1.2 Lingkungan <i>Software</i> .....	45
5.2 Batasan Implementasi .....	45
5.3 Implementasi Aplikasi.....	46
5.3.1 <i>Pre-Processing</i> .....	46
5.3.2 <i>Color Channel</i> .....	47
5.3.3 <i>Gray Level CO-OCCURENCE MATRIX (GLCM)</i> .....	50
<b>BAB 6 PENGUJIAN DAN ANALISIS.....</b>	<b>54</b>
6.1 Skenario Pengujian dan Analisis .....	54
6.1.1 Pengujian Nilai <i>k</i> .....	56
6.1.2 Pengujian Ekstraksi Fitur .....	57
6.1.3 Pengujian Perhitungan Jarak Tetangga Terdekat .....	59
<b>BAB 7 PENUTUP .....</b>	<b>62</b>
7.1 Kesimpulan.....	62

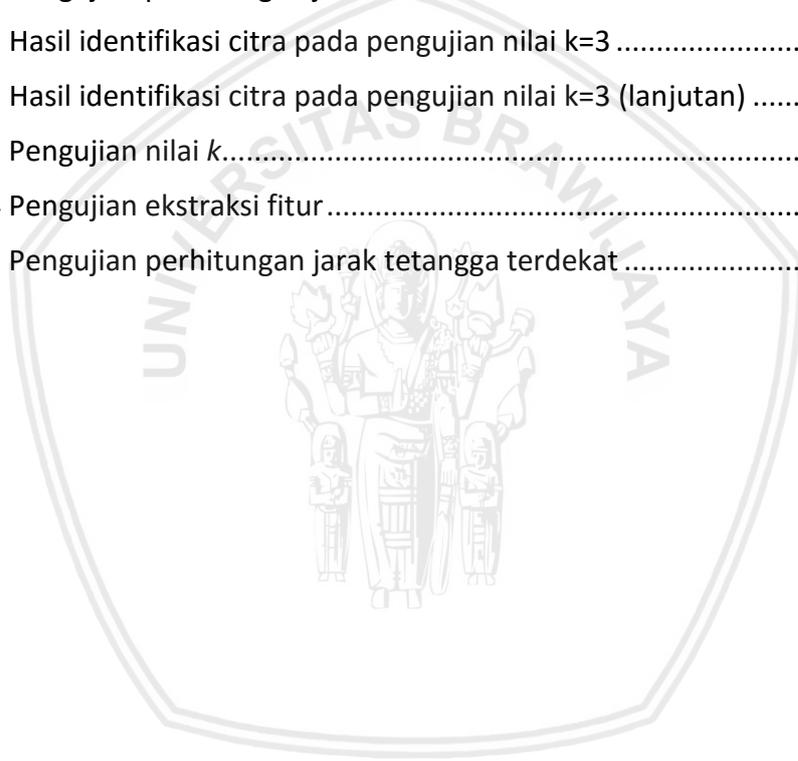


7.2 Saran .....	62
DAFTAR REFERENSI .....	63
LAMPIRAN A HASIL PENGUJIAN .....	66



## DAFTAR TABEL

Tabel 2.1 Kajian Pustaka .....	6
Tabel 4.1 Tabel Ekstraksi Fitur Tesktur dan Warna Citra Makanan.....	41
Tabel 4.2 Tabel Perbandingan Jarak Antar Kategori Kelas.....	42
Tabel 4.3 Tabel Citra Makanan dengan Jarak Terdekat dengan Citra Kueri.....	43
Tabel 4.4 Pengujian nilai k pada data seimbang dan data tidak seimbang .....	43
Tabel 4.5 Pengujian ekstraksi fitur .....	44
Tabel 4.6 Pengujian perhitungan jarak terdekat .....	44
Tabel 6.1 Hasil identifikasi citra pada pengujian nilai k=3 .....	54
Tabel 6.2 Hasil identifikasi citra pada pengujian nilai k=3 (lanjutan) .....	55
Tabel 6.3 Pengujian nilai k.....	56
Tabel 6.4 Pengujian ekstraksi fitur .....	58
Tabel 6.5 Pengujian perhitungan jarak tetangga terdekat .....	59



## DAFTAR GAMBAR

Gambar 2.1 Grafik representasi fungsi Gaussian Blur 2 dimensi.....	10
Gambar 2.2 (a) Gambar asli (b) Gambar hasil proses <i>opening</i> .....	12
Gambar 2.3 Bagan ilustrasi tahapan algoritme <i>k-fold cross validation</i> .....	15
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	17
Gambar 4.1 Diagram Alir Sistem .....	20
Gambar 4.2 Diagram Alir <i>Pre-Processing</i> .....	22
Gambar 4.3 Diagram Alir Ekstraksi Tesktur GLCM.....	22
Gambar 4.4 Diagram alir perhitungan matriks GLCM .....	23
Gambar 4.5 Diagram Alir Ekstraksi <i>Color Channel</i> .....	25
Gambar 4.6 Diagram alir perhitungan rata-rata pada citra tersegmentasi.....	25
Gambar 4.7 Diagram alir perhitungan <i>skewness</i> pada citra tersegmentasi.....	26
Gambar 4.8 Diagram alir perhitungan standar deviasi pada citra tersegmentasi	27
Gambar 4.9 Sampel citra makanan berukuran 3x3 piksel .....	28
Gambar 4.10 Nilai piksel <i>greyscale</i> pada sampel citra tersegmentasi .....	28
Gambar 4.11 Citra asli tersegmentasi berukuran 200x200 .....	28
Gambar 4.12 Gambar matriks <i>co-occurence</i> piksel (35,30) dan $d = 1$ sudut $0^\circ$ ...	29
Gambar 4.13 Tabel <i>co-occurence</i> GLCM untuk Sudut $0^\circ$ dari Gambar 4.18 .....	29
Gambar 4.14 Gambar matriks <i>co-occurence</i> piksel (34,30) dan $d = 1$ sudut $45^\circ$ .	29
Gambar 4.15 Tabel <i>co-occurence</i> GLCM untuk Sudut $45^\circ$ dari Gambar 4.20 .....	30
Gambar 4.16 Gambar matriks <i>co-occurence</i> piksel (35,34) dan $d = 1$ sudut $90^\circ$ .	30
Gambar 4.17 Tabel <i>co-occurence</i> GLCM untuk Sudut $90^\circ$ dari Gambar 4.22 .....	30
Gambar 4.18 Gambar matriks <i>co-occurence</i> piksel (30,33) dan $d = 1$ sudut $135^\circ$	30
Gambar 4.19 Tabel <i>co-occurence</i> GLCM untuk Sudut $135^\circ$ dari Gambar 4.24 ....	31
Gambar 4.20 Nilai piksel RGB citra sampel tersegmentasi berukuran 3x3 .....	39
Gambar 6.1 Grafik hasil pengujian nilai $k$ .....	57
Gambar 6.2 Grafik hasil pengujian ekstraksi fitur.....	58
Gambar 6.3 Grafik hasil pengujian perhitungan jarak pada data seimbang .....	60
Gambar 6.4 Grafik hasil pengujian perhitungan jarak pada data tidak seimbang .....	61

## DAFTAR KODE PROGRAM

Kode Program 5.1 Implementasi fungsi preprocessing().....	46
Kode Program 5.2 Implementasi fungsi mean() .....	47
Kode Program 5.3 Implementasi fungsi skewness() .....	48
Kode Program 5.4 Implementasi fungsi stdev().....	49
Kode Program 5.5 Implementasi matriks glcm() .....	51
Kode Program 5.6 Implementasi fungsi ekstraksifitur() .....	52



## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Dewasa ini mendeteksi dan mengklasifikasikan objek citra secara digital khususnya pada produk makanan sudah menjadi penelitian yang populer dan banyak dikembangkan dengan menggunakan sejumlah metode atau pendekatan. Fenomena tersebut dapat menjadi dasar dilakukannya suatu penelitian tentang pengenalan citra jenis makanan berdasarkan tekstur dan warnanya. Seperti yang kita ketahui bahwa makanan memiliki variasi tekstur dan warna yang dapat membedakan makanan satu dengan makanan yang lain. Variasi tekstur dan warna ini dapat digunakan sebagai salah satu komponen untuk meningkatkan performa dalam mengidentifikasi jenis-jenis makanan (Hoashi, et al., 2016). Secara manual, manusia dapat dengan mudah mengenali dan membedakan jenis-jenis makanan. Akan tetapi sistem penglihatan manusia tentu memiliki keterbatasan sehingga identifikasi jenis makanan yang dihasilkan berbeda-beda dan tidak konsisten antara manusia satu dengan manusia yang lain. Penelitian menunjukkan bahwa sistem penglihatan manusia dipengaruhi oleh beberapa faktor yaitu faktor keterbatasan visual manusia seperti usia dan penyakit mata serta faktor kondisi psikis manusia (Andriyanto, 2009).

Proses identifikasi makanan dapat dilakukan dengan membedakan fitur atau ciri-ciri yang ada pada makanan. Pada penelitian ini, penulis menggunakan warna dan tekstur sebagai fitur utama untuk mengetahui karakteristik makanan. Warna dan tekstur merupakan dua dari tiga karakteristik dasar citra yang banyak memberikan informasi agar mata manusia dapat dengan mudah membaca dan memahami kandungan informasi dalam sebuah citra (Sarifuddin, 2015). Selain itu, warna dan tektur juga merupakan fundamental karakteristik yang dimiliki citra secara alami dan menjadi bagian penting dalam memberikan informasi persepsi secara visual (Arivazhagan, et al., 2010).

Berbeda dengan mata manusia, proses identifikasi citra pada komputer tidak dapat dilakukan tanpa menggunakan algoritme analisis dan pengenalan sehingga dibutuhkan proses segmentasi pada citra. Proses segmentasi citra pada penelitian ini didasarkan atas warna dan tekstur yang masing-masing menggunakan metode *color channel* dan *Gray Level Co-Occurence Matrix* (GLCM). Warna adalah fitur visual yang paling sering digunakan dalam pencarian citra berbasis konten (Muller, et al., 2004). Sebuah citra digital adalah sebuah set dari piksel-piksel yang mana tiap piksel merepresentasikan warna. Model warna (*color model*) atau *color channel* adalah sebuah cara untuk merepresentasikan warna yang diindera manusia dalam komputasi (Agaputra, Wardani, & Siswanto, 2013). Terdapat tiga hal yang digunakan sebagai parameter yang merepresentasikan warna pada suatu citra yaitu *mean*, *standard deviation*, dan *skewness*. Pada penelitian sebelumnya, metode *color channel* dan *Gray Level Co-Occurence Matrix* (GLCM) digunakan untuk mengidentifikasi jenis buah. Hasil pengujian menunjukkan sebanyak 1326 jenis buah berhasil diidentifikasi dengan akurasi 86% (Arivazhagan, et al., 2010).

Rekaman citra terhadap objek-objek real di alam tidak selalu menunjukkan region-region dengan intensitas atau warna yang uniform atau homogen misalnya, citra pada kulit jeruk. Sekalipun warnanya seolah tampak homogen, namun karena memiliki permukaan berbintik dengan posisi acak maka intensitas cahaya yang dipantulkan menjadi berbeda. Permukaan berbintik pada kasus kulit jeruk tersebut dikategorikan sebagai fitur tekstur pada objek citra (Sarifuddin, 2015). Fitur tekstur menjadi komponen sekunder yang penting dalam pengklasifikasi data berbasis citra selain fitur warna. Penelitian membuktikan bahwa proses klasifikasi dengan menggunakan kombinasi fitur warna dan tekstur merupakan kombinasi yang sangat baik dibandingkan dengan hanya menggunakan salah satu fitur yaitu fitur warna atau fitur tekstur (Karegowda, Samreen, & Pushpalatha, 2015).

Pada penelitian ini, algoritme yang digunakan dalam proses segmentasi fitur tekstur pada objek citra adalah algoritme *Gray Level Co-Occurrence Matrix* (GLCM). *Gray level Co-Occurrence Matrix* (GLCM) adalah tabulasi mengenai frekuensi atau seberapa sering kombinasi nilai kecerahan piksel (*gray level*) yang berbeda posisinya muncul pada suatu objek citra (Anami & Burkpalli, 2009). Beberapa penelitian sebelumnya terkait penggunaan metode GLCM ini diantaranya adalah identifikasi makanan yang mengandung gula yang menghasilkan akurasi sebesar 90% (Anami & Burkpalli, 2009), penelitian tentang analisis dan pengenalan 85 kategori makanan yang menghasilkan akurasi sebesar 62,52% (Hajime, Taichi, & Keiji, 2016), dan penelitian oleh Neneng dan Fernando (2017) tentang klasifikasi jenis daging dengan akurasi sebesar 75,6%.

Pada proses selanjutnya, setelah berhasil mengidentifikasi nilai pada fitur warna dan tekstur pada citra makanan, penulis menggunakan *K-Nearest Neighbor* (KNN) *Classifier* untuk mengklasifikasikan fitur-fitur tersebut sesuai kategori makanan yang telah ditentukan. *K-Nearest Neighbor* (K-NN) *Classifier* merupakan metode yang sudah banyak digunakan secara luas. Dengan menggunakan informasi fitur-fitur pada data uji, KNN dapat mengklasifikasikan label data berdasarkan nilai k yang terdekat dengan data latih (Ye He, et al., 2014). Metode K-NN ini dipilih karena merupakan metode klasifikasi yang sederhana dan mudah untuk diimplementasikan dengan hasil klasifikasi yang baik. Selain itu, keakurasian metode ini sangat dipengaruhi oleh parameter k (Shaoyu, et al., 2014). *K-Nearest Neighbor* (KNN) *Classifier* banyak digunakan untuk penelitian terkait klasifikasi jenis makanan seperti penelitian yang dilakukan oleh Shaoyu, dkk (2014) yaitu identifikasi dan klasifikasi citra makanan cepat saji yang menghasilkan akurasi sebesar 74,23%. Penelitian lain oleh Ye He, dkk (2014) terkait klasifikasi jenis makanan menunjukkan dengan menggunakan 1453 data uji hasil akurasi yang didapat sebesar 81%.

Berdasarkan paparan di atas, maka diusulkan untuk menggunakan ekstraksi fitur *color channel* dan *Gray Level Co-Occurrence Matrix* (GLCM) sebagai algoritme yang menghasilkan nilai fitur warna dan tekstur pada citra makanan tersegmentasi serta *K-NN Classifier* sebagai metode untuk mengidentifikasi dan mengklasifikasi objek citra kueri. Dengan demikian, judul penelitian ini yaitu "Pengenalan Citra Jenis Makanan Menggunakan Ekstraksi Fitur *Color Channel*

dan *Gray Level Co-Occurrence Matrix*” yang diharapkan dapat memberikan informasi mengenai pengenalan jenis citra makanan berdasarkan tekstur dan warnanya.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan di muka, terdapat rumusan masalah yang dapat dijadikan dasar untuk membuat suatu teknik pengenalan citra jenis makanan sehat untuk program diet menggunakan ekstraksi fitur *color channel* dan *gray level Co-Occurrence Matrix* yaitu:

1. Bagaimana hasil evaluasi pengenalan citra jenis makanan yang menerapkan ekstraksi fitur *color channel* dan *gray level co-occurrence matrix*?
2. Bagaimana pengaruh ekstraksi fitur *color channel* dan *gray level co-occurrence matrix* terhadap hasil akurasi?

## 1.3 Tujuan

Tujuan yang dicapai adalah sebagai berikut:

1. Mengukur tingkat akurasi pengenalan citra jenis makanan yang menerapkan ekstraksi fitur *color channel* dan *gray level co-occurrence matrix*.
2. Mengetahui pengaruh ekstraksi fitur *color channel* dan *gray level co-occurrence matrix* terhadap hasil akurasi.

## 1.4 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut:

1. Mendapatkan hasil akurasi pengenalan citra jenis makanan yang menerapkan ekstraksi fitur *color channel* dan *gray level co-occurrence matrix*.
2. Mendapatkan analisis pengaruh penggunaan ekstraksi fitur *color channel* dan *gray level co-occurrence matrix* dalam pengenalan citra jenis makanan.

## 1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Terdapat 15 jenis citra yang digunakan sebagai dataset yaitu donat, mie instan keriting, telur ceplok, rendang, tomat, buah seperti stroberi, pisang hijau, pisang kuning, jeruk oranye, dan jeruk hijau, serta *snack* berupa Beng-Beng, Gerry Salut, Milo Nuggets, Biskuit Vanila, dan Genji Pie.
2. Teknik pengambilan dataset dilakukan dengan cara memfoto makanan sebanyak tiga kali berdasarkan 3 level ketinggian dan 3 jenis kemiringan.
3. Satu jenis makanan dilakukan pengambilan foto sebelum dan sesudah dimakan yang diukur berdasarkan 3 ukuran yaitu  $\frac{1}{4}$  bagian,  $\frac{1}{2}$  bagian, dan  $\frac{3}{4}$  bagian.
4. Bahasa pemrograman yang digunakan dalam penelitian ini adalah Python.

## 1.6 Sistematika Pembahasan

Sistematika pembahasan pada skripsi ini merujuk pada suatu sistematika sebagai gambaran dan penjabaran secara menyeluruh yang meliputi beberapa bab berikut ini:

### **BAB 1 PENDAHULUAN**

Bab pendahuluan akan membahas tentang latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat skripsi ini dibuat, serta sistematika pembahasan penelitian.

### **BAB 2 LANDASAN KEPUTUSAN**

Pada bab ini akan dibahas mengenai dasar teori pendukung maupun referensi yang menjadi dasar pengenalan citra jenis makanan yang menerapkan ekstraksi fitur *Color Channel* dan *Gray Level Co-Occurrence Matrix* dengan metode *K-Nearest Neighbor*. Teori-teori yang digunakan yaitu sistem pengenalan citra jenis makanan, *Gaussian Blur*, transformasi RGB ke HSV, *grayscale*, transformasi biner, *opening*, ruang warna (*color channel*), *Euclidean distance*, *Gray Level Co-Occurrence Matrix* (GLCM), dan *K-Nearest Neighbor* (K-NN).

### **BAB 3 METODOLOGI PENELITIAN**

Pada bab metodologi ini akan dibahas mengenai langkah-langkah serta penjabaran metode yang digunakan dalam penelitian seperti kepustakaan, pengumpulan data, analisis kebutuhan, menentukan metode/algorithm, perancangan algoritme, implementasi, pengujian dan analisis, dan kesimpulan.

### **BAB 4 ANALISIS DAN PERANCANGAN**

Pada bab analisis dan perancangan ini akan dibahas mengenai perancangan sistem yang akan digunakan dalam proses mengenali atau mengidentifikasi jenis makanan sebagai panduan dalam pengimplementasian sistem yang akan dibangun.

### **BAB 5 IMPLEMENTASI**

Pada bab ini akan dilakukan implementasi algoritme ekstraksi fitur warna yaitu *color channel* dan ekstraksi fitur tekstur yaitu *Gray Level Co-Occurrence Matrix* (GLCM) menggunakan metode *K-Nearest Neighbor* (K-NN) sesuai dengan perancangan sistem pada Bab 4.

### **BAB 6 PENGUJIAN DAN ANALISIS**

Pada bab ini akan menjelaskan tentang hasil pengujian disertai analisis pada proses implementasi sistem yang menggunakan ekstraksi fitur warna yaitu *color channel* dan ekstraksi fitur tekstur yaitu *Gray Level Co-Occurrence Matrix* (GLCM) menggunakan metode *K-Nearest Neighbor* (K-NN).

### **BAB 7 PENUTUP**

Pada bab terakhir ini akan dibahas tentang kesimpulan hasil analisis proses implementasi sistem pengenalan citra jenis makanan menggunakan metode

*color channel* dan GLCM yang kemudian diklasifikasi dengan K-NN, hasil pengukuran akurasi sistem, dan saran-saran dari penulis tentang pengembangan sistem.



## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Penelitian sebelumnya terkait metode ekstraksi dan metode klasifikasi yang digunakan banyak memiliki kesamaan dengan metode-metode yang ada pada penelitian ini. Tabel 2.1 berikut akan menjabarkan tentang penelitian-penelitian terdahulu yang memiliki kesamaan metode dan objek yang dapat digunakan sebagai acuan bagi penulis dalam menyelesaikan permasalahan terkait pengenalan citra jenis makanan.

Tabel 2.1 Kajian Pustaka

No	Judul dan Objek	Metode	Hasil
1.	<p><b>Judul:</b> <i>Texture Based Identification and Classification of Bulk Sugary Food Objects</i></p> <p><b>Objek:</b> Masukan sistem berupa citra makanan manis yang berasal dari India Selatan seperti <i>Applecake, Bundeladu, Burfi, Doodhpeda, Jamun, Jilebi, Kalakand Ladakiladu, Mysorepak,</i> dan <i>Suraliholige</i>. Makanan manis tersebut sekilas terlihat sama ketika dipajang di etalase toko, namun yang membedakannya adalah tekstur sehingga pada penelitian ini fitur yang digunakan berupa tekstur menggunakan metode GLCM. Klasifikasi makanan dilakukan dengan <i>JST Feed Forward</i>.</p>	<p>1. Citra sampel dengan komponen warna RGB dilakukan perhitungan ekstraksi fitur tesktur (GLCM) masing-masing 4 arah yaitu 0°, 45°, 90°, dan 135° dengan nilai <math>d = 1</math> yang nilainya tergantung pada nilai arah <math>\varphi</math></p> <p>2. Menghitung matriks <i>co-occurence</i> yang nilainya tidak bergantung pada arah dengan menggunakan persamaan berikut:</p> $C = \frac{1}{4} (P_0^\circ + P_{45^\circ} + P_{90^\circ} + P_{135^\circ})$ <p>3. Menghasilkan output berupa 14 fitur GLCM yaitu Contrast, Correlation, Entrophy, Energy, Homogeneity, Dissimilarity, Smoothness, Cluster Shade, Cluster Performance, Angular Second Moment, Third</p>	<p>Hasil penelitian menunjukkan bahwa penggunaan metode GLCM sebagai fitur tekstur sangat efisien. Total efisiensi dan akurasi metodologi pada citra jenis makanan manis ini sebesar 90%.</p>



	<p>Penelitian ini bekerja untuk dapat mengawasi dan menghidangkan makanan yang ada di restoran, hotel, dan pusat perbelanjaan secara otomatis dengan menggunakan robot</p> <p>(Anami &amp; Burkpalli, 2009)</p>	<p>Moment, Mean, Variance, Starndard Deviation, Dan Maximum Probability.</p> <p>4. Selanjutnya nilai ekstraksi fitur testur tersebut digunakan untuk melatih model JST yang sudah dibangun yang nantinya akan digunakan untuk mengklasifikasikan sampel makanan manis.</p>	
2.	<p><b>Judul:</b> <i>Fruit Recognition Using Color and Texture Feature</i></p> <p><b>Objek:</b> Objek penelitian yang digunakan adalah buah yang ada di <i>Supermarket</i> yang terdiri 15 kategori buah dengan total 2633 citra buah. Citra tersebut kemudian dibagi menjadi 2 data yaitu 50% sebagai data latih, sisanya sebagai data uji. Tujuan dari penelitian ini adalah untuk mengukur efisiensi perpaduan antara fitur warna dan tekstur dalam sistem pengenalan citra buah.</p> <p>(Arivazhagan, Shebiah, &amp; Ganesan, 2010)</p>	<p>1. Citra buah dengan komponen warna RGB dikonversikan menjadi HSV</p> <p>2. Citra tersebut kemudian diekstraksikan dengan GLCM untuk menghasilkan 5 fitur yaitu <i>Contrast, Energy, Local Homogeneity, Cluster Shade, Dan Cluster Prominence</i>, sedangkan ekstrasi warna dengan color channel untk menghasilkan 4 nilai fitur yaitu <i>Mean, Standard Deviation, Skewness</i>, dan <i>Kurtosis</i> yang diperoleh dari komponen warna H dan S. Total terdapat 13 nilai fitur yang digunakan untuk 1 citra buah yang</p>	<p>Penelitian ini berhasil mengukur akurasi dari perpaduan fitur warna dan tekstur untuk mengidentifikasi jenis buah. Hasil eksperimen menunjukkan akurasi sistem dengan menggunakan perpaduan 2 fitur yaitu fitur warna dan tekstur lebih besar daripada sistem yang hanya menggunakan salah satu dari dua fitur tersebut. Sistem dengan penggunaan 2 fitur memiliki akurasi 86%, sedangkan sistem dengan fitur warna atau tektur saja menghasilkan akurasi masing-</p>

		<p>disimpan dalam <i>database</i> fitur</p> <p>3. Berdasarkan ke-13 fitur tersebut, identifikasi citra buah dilakukan dengan mengklasifikasikan nilai fitur ke dalam 15 kategori buah dengan metode <i>Minimum Distance Classifier</i>.</p>	<p>masing sebesar 46% dan 71%.</p>
3.	<p><b>Judul:</b> <i>Color and Texture Based Identification and Classification of Food Grains Using Different Color Model and Haralick Features</i></p> <p><b>Objek:</b> Eksperimen pada penelitian ini menggunakan 10 kategori kelas untuk citra biji-bijian. Proses identifikasi citra dilakukan dalam 2 tahap; tahap pertama yaitu menghitung nilai fitur, sedangkan tahap kedua yaitu proses klasifikasi citra biji-bijian berdasarkan nilai fitur yang sesuai dengan kategori kelas yang sudah ditentukan sebelumnya. Citra asli difoto di bawah sinar matahari kemudian dilakukan proses <i>resized</i> menjadi ukuran 1024x1024. Pengklasifikasian dilakukan dengan</p>	<p>Perhitungan Nilai Fitur</p> <ol style="list-style-type: none"> <li>1. Membaca citra masukan dengan komponen warna RGB yang kemudian diubah ke komponen warna yang sesuai</li> <li>2. Pembentukan matriks <i>co-occurrence</i> dan perhitungan 5 fitur <i>Haralick</i>.</li> <li>3. Perhitungan <i>Mean</i> dan <i>Standard Deviation</i> menggunakan nilai <i>Chrominance</i></li> <li>4. Pembentukan histogram secara kumulatif untuk mengekstraksi fitur global yaitu mean, standard deviation, dan slope pada garis persamaan regresi linier</li> <li>5. Menyimpan hasil ekstraksi 12 nilai fitur ke dalam <i>database</i></li> </ol> <p>Algoritme Klasifikasi:</p> <ol style="list-style-type: none"> <li>1. Membaca citra uji</li> <li>2. Mengonversikan komponen warna</li> </ol>	<p>Penelitian ini menggunakan 3 jenis tipe warna yaitu L*a*b, HSV, dan HSI serta K-NN sebagai metode klasifikasinya. Hasil pengukuran akurasi berdasarkan tipe warnanya adalah sebagai berikut:</p> <ol style="list-style-type: none"> <li>1. Eksperimen pada tipe warna L*a*b dengan penggunaan data uji sebesar 18,75% untuk K=7 memiliki akurasi 95,72%.</li> <li>2. Eksperimen pada tipe warna L*a*b dengan penggunaan data uji sebesar 12,5% untuk K=5 memiliki akurasi 94,15%.</li> <li>3. Eksperimen pada tipe warna HSV dengan penggunaan data uji sebesar</li> </ol>



	<p>menggunakan K-NN Classifier dan Minimum Distance Classifier (Patil, Malemath, &amp; Yadahalli, 2011)</p>	<p>pada citra uji dan menghitung nilai fiturnya</p> <ol style="list-style-type: none"> <li>3. Menentukan nilai k (k=1,3,5,7)</li> <li>4. Menentukan database untuk menyimpan hasil perhitungan nilai fitur</li> <li>5. Menghitung jarak nilai fitur antara citra uji dan citra pada basis data</li> <li>6. Mengurutkan hasil perhitungan jarak dari jarak terkecil hingga jarak terbesar</li> <li>7. Jarak minimal yang digunakan dalam proses klasifikasi adalah k=1 dan K-NN Classifier biasanya menggunakan nilai k=3, 5, dan 7 untuk menentukan kategori kelas pada citra yang diidentifikasi</li> </ol>	<p>25% untuk K=5 memiliki akurasi 88,02%.</p> <ol style="list-style-type: none"> <li>4. Eksperimen pada tipe warna HSV dengan penggunaan data uji sebesar 12,5% untuk K=5 memiliki akurasi 85,17%.</li> <li>5. Eksperimen pada tipe warna HSI dengan penggunaan data uji sebesar 25% untuk K=3 memiliki akurasi 89,21%.</li> <li>6. Eksperimen pada tipe warna HSI dengan penggunaan data uji sebesar 12,5% untuk K=5 memiliki akurasi 84,46%.</li> </ol>
--	---	--	---

## 2.2 Pengolahan Citra Digital

Definisi dari pengolahan citra digital merupakan suatu algoritme yang diimplemetasikan dalam suatu teknologi untuk memroses citra secara digital. Hasil dari pengolahan suatu citra secara digital sendiri berupa gambar atau nilai ekstraksi dari fitur representasi yang menjadi karakteristik atau ciri dari gambar asli. Prose mengola citra secara digital suda banyak diimplementasikan dalam berbagai aspek yaitu *artificial intelligent system (AI system)*, citra medis, pengendalian jarak jauh, fotografi, dan forensik (Zhou, et al., 2010).

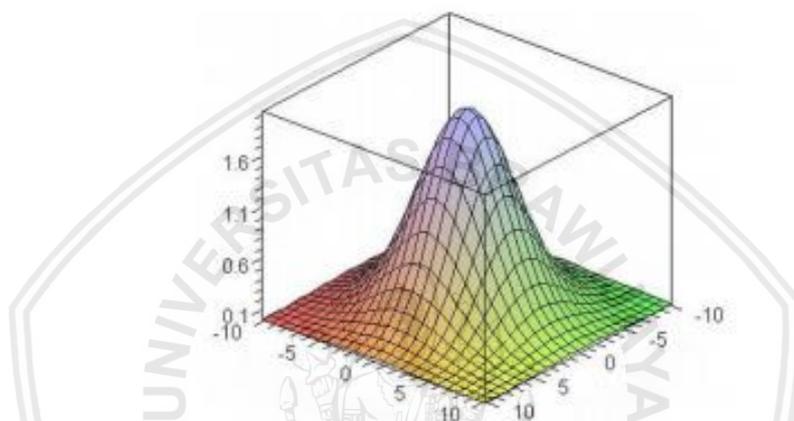
### 2.2.1 Segmentasi Citra

Segmentasi citra merupakan salah satu metode yang paling sering digunakan untuk mengklasifikasi piksel pada citra. Pada prosesnya, segmentasi citra dilakukan dengan mempartisi citra ke dalam kelompok area secara diskrit yang memiliki kesamaan tingkat kontras yang tinggi baik di dalam kelompok area maupun antar kelompok area (Dhanachandra, et al., 2015).



### 2.2.1.1 Gaussian Blur

Proses blurring pada gambar dengan menggunakan fungsi Gaussian disebut *Gaussian Blurring*. Secara umum, metode ini digunakan untuk mengurangi noise dan detail gambar pada pengolahan citra digital. Konsep dari metode ini adalah memperbaiki nilai piksel dengan melakukan perhitungan berat rata-rata piksel tetangganya. Apabila nilai bobot rata-rata piksel tetangga tersebut lebih besar daripada nilai rerata pada *blur pixel* nya, maka hal tersebut menunjukkan bahwa nilai rerata bobot pada setiap piksel memiliki nilai yang sangat besar. Pada intinya, Metode *Gaussian Blur* merupakan metode yang membantu mencari nilai bobot setiap piksel tetangga. Fungsi *Gaussian Blur* dalam citra 2 dimensi ditunjukkan pada Gambar 2.1



**Gambar 2.1 Grafik representasi fungsi Gaussian Blur 2 dimensi**

Sumber: (Bozkurt, et al., 2015)

Berdasarkan grafik tersebut dapat dilihat bahwa *blur pixel* memiliki puncak jika dibandingkan dengan nilai bobot rerata piksel tetangga lainnya. Selain itu, jarak maksimum piksel pada Gambar 2.1 di atas adalah lima piksel yang masing-masing jarak memiliki ratio piksel yang sama. Dengan demikian, dapat disimpulkan bahwa ukuran window adalah sepuluh yang juga disebut sebagai ukuran kernel (Bozkurt, et al., 2015).

### 2.2.1.2 Transformasi RGB menjadi HSV

Ruang warna RGB adalah ruang warna yang terdiri atas tiga warna utama yaitu *Red*, *Green*, dan *Blue*. Dewasa ini model warna RGB telah banyak digunakan pada banyak aplikasi seperti kamera digital, layar digital, mesin pemindai warna (*scanner*) dan gambar digital. Dalam proses segmentasi citra berwarna, ruang warna RGB merupakan ruang warna yang ambivalen. Di sisi lain, ruang warna RGB akan sangat menyulitkan jika digunakan pada proses segmentasi gambar, namun secara visual ruang warna ini merupakan ruang warna yang paling sesuai dalam pemrosesan biologis warna pada manusia (Loesdau, et al., 2014). Ruang warna HSV yang terdiri atas parameter *Hue*, *Saturation*, dan *Value* menjadi salah satu warna yang sesuai dalam proses pengolahan citra digital karena mampu memisahkan informasi warna sesuai dengan sistem penglihatan manusia (Afirianto & Amalia, 2016).

### 2.2.1.3 Grayscale

Citra keabuan atau disebut juga dengan citra *grayscale* merupakan sebuah citra yang intensitas warnanya direpresentasikan dalam 2 bit yaitu 0 dan 1. Nilai 0 merepresentasikan hitam, sedangkan angka 1 merepresentasikan warna putih. Dalam citra RGB kedua bit tersebut direpresentasikan dengan nilai 0 sebagai warna hitam dan 255 sebagai warna putih. Hal ini disebabkan karena dalam citra RGB nilai maksimal suatu piksel adalah 255 sedangkan nilai minimal piksel adalah 0, semakin tinggi nilai piksel maka warna citra akan semakin terang (Kadir & Susanto, 2013). Citra dengan skala keabuan didapatkan dengan melakukan konversi nilai piksel dari RGB ke *grayscale* melalui persamaan 2.1 berikut.

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

dimana R, G, dan B merupakan nilai untuk piksel *Red*, *Green*, dan *Blue*

### 2.2.1.4 Thresholding

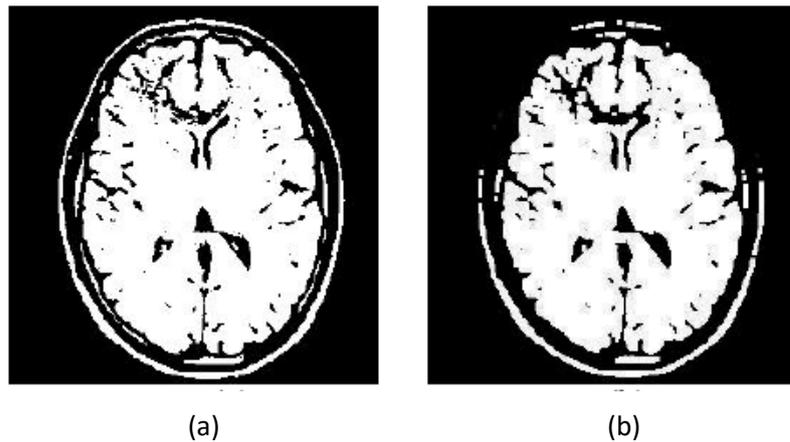
*Thresholding* merupakan metode segmentasi gambar yang paling sederhana. Dari citra skala abu-abu, *thresholding* dapat digunakan untuk mengonversikan citra menjadi citra biner. Proses segmentasi yang menetapkan setiap piksel dalam dua kelas ini menggunakan minimal satu parameter nilai skala warna untuk mendefinisikan batasnya. Keuntungan dari penggunaan metode ini adalah mengurangi kerumitan data dan menyederhanakan proses pengenalan dan klasifikasi citra. Masukan data pada proses *thresholding* ini berupa citra *grayscale*. Hasil dari proses ini berupa citra biner sebagai citra yang sudah tersegmentasi. Jika dilihat secara kasat mata, warna hitam pada citra biner merupakan latar belakang gambar yang bernilai 0, sedangkan latar depan berwarna putih yang bernilai 1. Secara matematis, hasil *thresholding* pada citra biner ditunjukkan pada persamaan 2.2 berikut.

$$g(x, y) = \begin{cases} 0 & f(x, y) < T \\ 1 & f(x, y) > T \end{cases} \quad (2.2)$$

Dimana 0  $f(x,y)$  merupakan *background pixel*, piksel dengan nilai di bawah ambang batas (T), sedangkan 1  $f(x,y)$  adalah *background pixel* dengan nilai piksel di atas nilai ambang batas (T) (Bhargavi & Jyothi, 2014).

### 2.2.1.5 Opening

Proses *opening* dalam segmentasi citra digital didukung dengan menggunakan operasi erosi kemudian diikuti oleh proses dilasi yang implementasinya dapat diterapkan untuk menghilangkan nilai-nilai piksel pada area tertentu yang nilainya terlalu kecil untuk dilakukan perataan elemen piksel. Hasil proses *opening* ini ditunjukkan pada Gambar 2.2. Pada gambar tersebut proses perataan elemen piksel sering menampilkan *probe*, proses ini dinamakan *probing*. Proses *probing* adalah proses pemeriksaan gambar dengan nilai piksel terkecil sehingga dapat difilter atau dihilangkan (N. & C., 2014).



Gambar 2.2 (a) Gambar asli (b) Gambar hasil proses *opening*  
 Sumber: (N. & C., 2014)

### 2.3 Ruang Warna (*Color Channel*)

Ruang warna didefinisikan sebagai suatu spesifikasi sistem koordinat dan suatu sub ruang dalam sistem tersebut dengan setiap warna dinyatakan dengan satu titik di dalamnya (Gonzales & Woods, 2002). Ruang warna digunakan sebagai alat untuk mengelompokkan warna-warna yang terstandarisasi. Terdapat beberapa jenis ruang warna diantaranya RGB, HSI, CMY, LUV, dan YIQ.

Menurut Kadir dan Susanto (2013), RGB merupakan ruang warna yang paling sering digunakan dalam perangkat komputer karena sesuai dengan watak manusia dalam menangkap warna. Penelitian ini menggunakan ruang warna RGB (*Red Green Blue*) karena merupakan warna dasar dan mudah untuk dilakukan proses analisis pada citra. Selain itu, dataset yang digunakan adalah citra yang menggunakan RGB sebagai komponen warnanya. Pada color channel terdapat 3 parameter nilai ekstraksi fitur yang digunakan yaitu *mean*, *skewness*, dan standar deviasi (Kodituwakku & Selvarajah, 2010). Perhitungan nilai rerata (*mean*) pada citra ditunjukkan pada persamaan 2.3 berikut.

$$mean = \sum_{i=1}^n \sum_{j=1}^m \frac{x_{ij}}{mn} \tag{2.3}$$

Keterangan:

n = jumlah data pada indeks ke-i

m = jumlah data pada indeks ke-j

$x_{ij}$  = nilai piksel untuk baris ke-i dan kolom ke-j

Persamaan nilai *skewness* pada citra ditunjukkan pada persamaan 2.4 berikut.

$$skewness = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M \left[ \frac{P(i,j) - \mu}{\sigma} \right]^3 \tag{2.4}$$

Dimana  $P(i,j)$  merupakan nilai piksel pada koordinat indeks pada sumbu (x,y),  $\sigma$  merupakan nilai dari standar deviasi.

Sedangkan perhitungan nilai standar deviasi pada citra ditunjukkan pada persamaan 2.5 berikut.

$$variance = \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m (x_{ij} - mean)^2 \tag{2.5}$$

$$stddev = \sqrt{variance} \quad (2.6)$$

## 2.4 Euclidean Distance

Dalam prosesnya, segmentasi warna pada suatu citra dilakukan dengan menentukan homogenitas warna serta memisahkan dua segmen warna yang memiliki intensitas warna yang berbeda berdasarkan acuan dari peta warna. Pemisahan dua segmen warna ini dilakukan dengan melakukan perhitungan perbedaan jarak antar kedua warna sebagai alat ukurnya. Menurut Madenda (2015), untuk ruang warna RGB dalam koordinat Kartesian 3-D, jarak warna antar dua piksel dapat dihitung dengan menggunakan rumus jarak *Euclidean* yang dinyatakan dalam persamaan berikut:

$$\Delta E = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \quad (2.7)$$

$$\Delta E = |R_1 - R_2| + |G_1 - G_2| + |B_1 - B_2| \quad (2.8)$$

## 2.5 Manhattan Distance

*Manhattan distance* atau disebut juga dengan *city blocked distance* merupakan suatu metode perhitungan jarak yang hasilnya didapatkan dengan cara melakukan kalkulasi jarak yang akan ditempuh dari satu titik data ke titik data yang lain sesuai dengan jalur yang ada pada grid. Secara matematis, jarak *manhattan* adalah jumlah dari perbedaan selisih antar data. Jarak *manhattan* ini juga dikenal dengan sebutan jarak L1. Jarak antara titik  $x = (x_1, x_2, x_3 \dots x_n)$  dan titik  $y = (y_1, y_2, y_3 \dots y_n)$  ditunjukkan pada persamaan 2.9 berikut.

$$MD_{(x,y)} = \sum_{i=1}^n |x_i - y_i| \quad (2.9)$$

Dimana  $n$  adalah jumlah variabel, dan  $x_i$  dan  $y_i$  adalah nilai dari variabel ke- $i$  pada titik  $x$  dan  $y$  (Ponnmoli & Selvamuthukumar, 2014).

## 2.6 Gray Level CO-OCCURENCE MATRIX (GLCM)

Gray Level Co -Occurance Matrix (GLCM) atau bisa disebut dengan matriks co-occurrence merupakan satu dari sejumlah metode yang paling sering digunakan dalam analisis tekstur. GLCM menggunakan perhitungan tekstur pada orde kedua, dimana pada orde pertama dilakukan perhitungan statistik yang didasarkan pada nilai piksel pada citra asli seperti varians, dan tidak memerhatikan hubungan antar piksel. Namun pada orde kedua, hubungan antar piksel diperhitungkan (Kadir dan Susanto, 2013). Menurut Haralick, et al (1973) disitasi dalam Mahardika, Sari, dan Dewi (2018), langkah-langkah ekstraksi fitur GLCM adalah sebagai berikut:

1. Menghitung nilai matriks kookurensi awal dengan menggunakan sudut  $0^\circ, 45^\circ, 90^\circ, 135^\circ$  dan  $d$  sebagai jarak piksel terhadap tetangga yang diinginkan. Namun biasanya menggunakan  $d = 1$ .
2. Membentuk matriks simetris dengan menjumlahkan nilai matriks kookurensi dengan transposenya.

3. Melakukan normalisasi terhadap nilai matriks agar tidak menyebabkan ketergantungan sehingga ketika dijumlahkan nilainya adalah 1.
4. Menghitung nilai ekstraksi fitur.

Terdapat 5 karakteristik pada GLCM yang biasa digunakan dalam ekstraksi tekstur. Berikut penjelasan karakteristik tersebut:

1. *Homogeneity, Angular Second Moment (ASM)*, yaitu *energy* yang digunakan untuk mengukur konsentrasi pasangan intensitas pada matriks *co-occurrence*.

$$Homogeneity = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{P(i,j)}{1+(i,j)^2} \quad (2.10)$$

2. *Contrast*, digunakan untuk mengukur variasi atau perbedaan intensitas tingkat keabuan dalam citra.

$$Contrast = \sum_{i=0}^G \sum_{j=0}^G |i-j|^2 P(i,j) \quad (2.11)$$

3. *Correlation*, digunakan untuk mengukur korelasi dan ketergantungan antara piksel dengan tingkat keabuan *i* dan piksel dengan tingkat keabuan *j* dalam citra.

$$Correlation = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{\{i \times j\} \times P(i,j) - \{\mu_x \times \mu_y\}}{\sigma_x \times \sigma_y} \quad (2.12)$$

4. *Sum of Square (Variance)*, digunakan untuk mengukur sebaran atau variasi nilai keabuan pada matriks kookurensi awal. Citra dengan sebaran derajat keabuan yang kecil akan menghasilkan variance yang kecil pula.

$$Variance = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i - \mu)^2 P(i,j) \quad (2.13)$$

5. *Entropy*, digunakan untuk mengukur tingkat ketidakaturan bentuk atau distribusi intensitas citra matriks *co-occurrence*.

$$Entropy = - \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P(i,j) \times \log(P(i,j)) \quad (2.14)$$

## 2.7 Metode *K-Nearest Neighbor (K-NN)*

Metode ini merupakan metode yang menggunakan algoritme *supervised learning* yang dapat mengklasifikasikan suatu data berdasarkan nilai *k* sebagai nilai yang menentukan suatu data untuk dapat masuk ke suatu kelompok baru sesuai dengan jumlah *k* yang diurutkan dari nilai paling kecil pada suatu data. Algoritme ini bertujuan untuk membentuk suatu kelompok data baru sesuai dengan atribut dan data latihnya.

Pada prosesnya, metode K-NN ini berusaha mencari jarak terpendek dari data kueri terhadap sampel data latih sesuai dengan jumlah nilai *k* yang sudah ditentukan. Kemudian mengambil data terbanyak dari *k* tetangga sehingga data kueri dapat dengan mudah diproyeksi. Penentuan nilai *k* secara optimal dapat dilakukan dengan menggunakan optimasi parameter seperti *k-fold cross validation* (Syafitri, 2008).



### 2.7.1 Konsep *K-Nearest Neighbor* (K-NN)

Metode K-NN ini memiliki konsep utama yaitu mencari jarak terpendek diantara dua variabel yaitu data uji dan jumlah nilai  $k$  tetangga terdekat pada data latih . Tahapan yang dilakukan pada metode K-NN adalah sebagai berikut (Priandana, et al., 2016):

1. Menentukan nilai  $k$ .
2. Menghitung jarak antara data pada citra kueri dan data pada setiap data latih.
3. Memperoleh nilai  $k$  terhadap data dengan jarak terdekat.
4. Nilai  $k$  merupakan jumlah data dengan jarak terdekat dengan kelas dominan.
5. Nilai dominan merupakan kelas terhadap data uji.

### 2.8 *K-Fold Cross Validation*

Dalam prosesnya, *k-fold cross validation* akan mengatur set data secara random kemudian dipartisi menjadi beberapa lipatan dengan ukuran yang sama. Lipatan atau partisi data set ini akan secara bergantian berubah sebagai set data uji hingga kelipatan akhir. Ilustrasi pergantian set data uji dengan penentuan nilai  $k=5$  ditunjukkan pada Gambar 2.3.



**Gambar 2.3** Bagan ilustrasi tahapan algoritme *k-fold cross validation*

Sumber: (Shaykh, 2018)

Selanjutnya, hasil akurasi pada setiap iterasi dihitung dan ditotal hingga berakhir pada iterasi kelima (apabila  $k=5$ , tergantung nilai  $k$ ). Perhitungan akurasi adalah jumlah semua data dengan pengklasifikasian benar dibagi dengan banyak data set. Dengan demikian, dapat kita ketahui bahwa metod ini bergantung pada banyaknya nilai  $k$  yang ditentukan di awal (Lu, s.l). Nilai  $k$  pada proses pengujian dengan *k-fold cross validation* ini tidak diambil secara acak, seperti proses perhitungan akurasi melainkan dipilih dengan terlebih dahulu memeriksa datanya. Secara umum, nilai  $k$ -tetangga terdekat yang besar lebih baik karena dapat menghilangkan *noise*, namun itu tetap tidak menjadi jaminan bahwa nilai akurasi yang dihasilkan akan besar juga. Penggunaan metode *k-fold cross validation* atau validasi silang ini merupakan cara lain untuk mendapatkan nilai  $k$ -

tetangga terdekat yang baik dengan menggunakan kumpulan data independen untuk melakukan validasi terhadap nilai  $k$ -tetangga terdekat (Ma, et al., 2014).

## 2.9 Akurasi

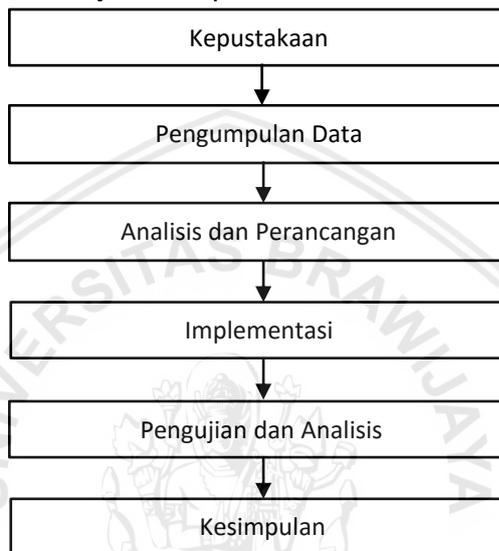
Metode perhitungan akurasi bertujuan untuk mengukur jumlah citra uji yang berhasil diklasifikasi dengan baik (*relevant*) kemudian dibagi dengan jumlah keseluruhan dataset. Nilai akurasi data pada penelitian ini direpresentasikan dalam bentuk persen (Qur'ania, et al., 2012). Secara matematis perhitungan akurasi ditunjukkan pada persamaan 2.15 berikut:

$$\text{Akurasi (A)} = \frac{\text{Jumlah klasifikasi yang benar}}{\text{Jumlah seluruh dataset}} \quad (2.15)$$



## BAB 3 METODOLOGI

Kerangka kerja merupakan langkah-langkah dan metode dalam melakukan penelitian. Suatu penelitian biasanya selalu dimulai dengan suatu perencanaan yang seksama yang mengikuti sejumlah petunjuk yang disusun secara sistematis sehingga hasilnya dapat mewakili kondisi yang sebenarnya dan dapat dipertanggung jawabkan (Eka & Edi, 2017). Bab ini menjelaskan kerangka kerja (metodologi) penelitian dengan jenis penelitian *Non-Implementatif Deskriptif* ini. Diagram tahapan penelitian dijelaskan pada Gambar 3.1



**Gambar 3.1 Diagram Alir Metodologi Penelitian**

### 3.1 Studi Literatur

Pada bab ini akan dibahas mengenai dasar teori pendukung maupun referensi yang menjadi dasar penelitian ini. Dasar teori yang dicantumkan antara lain sebagai berikut:

1. Kajian Pustaka
2. Pengolahan Citra Digital
3. *Gaussian Blur*
4. Tranformasi RGB menjadi HSV
5. *Grayscaleing*
6. *Thresholding*
7. *Opening*
8. Ruang Warna (*Color Channel*)
9. *Euclidean Distance*
10. *Manhattan Distance*
11. *Gray Level CO-OCCURENCE MATRIX (GLCM)*
12. Metode *K-Nearest Neighbor (K-NN)*
13. *Konsep K-Nearest Neighbor (K-NN)*
14. *K-Fold Cross Validation*
15. Akurasi

### 3.2 Pengumpulan Data

Pengumpulan data dilakukan dengan memfoto 15 jenis objek citra yang memiliki latar belakang (*backgorund*) berwarna putih. Proses pengambilan 15 dataset ini menggunakan kamera dari *handphone* Samsung Galaxy J7 Prime 8 MP. Teknik pengambilan dataset dilakukan dengan cara mengambil foto makanan sebanyak tiga kali berdasarkan 3 level ketinggian dan 3 jenis kemiringan. Satu jenis makanan dilakukan pengambilan foto sebelum dan sesudah dimakan yang diukur berdasarkan 3 ukuran yaitu  $\frac{1}{4}$  bagian,  $\frac{1}{2}$  bagian, dan  $\frac{3}{4}$  bagian. Total citra yang diambil adalah 450 citra yang terdiri atas citra berukuran  $\frac{1}{4}$  bagian,  $\frac{1}{2}$  bagian, dan  $\frac{3}{4}$  bagian masing-masing diambil sebanyak tiga foto.

Citra dataset tersebut merupakan citra utama yang nantinya digunakan sebagai citra pustaka untuk selanjutnya dilakukan *pre-processing* sehingga didapatlah citra dengan fokus objek tanpa adanya latar belakang. Citra hasil *pre-processing* ini selanjutnya diekstraksi berdasarkan 2 jenis fitur yaitu fitur warna dengan analisis ruang warna RGB dan fitur tekstur dengan analisis menggunakan metode GLCM. Setelah melewati proses ekstraksi, maka citra dataset akan menghasilkan nilai fitur warna dan tekstur yang kemudian disimpan sebagai database citra atau citra pustaka. Database ini nantinya akan menjadi informasi bagi citra query yang dilihat berdasarkan kecocokan nilai fitur warna dan tekstur.

### 3.3 Analisis dan Perancangan

Setelah berhasil membuat database untuk citra pustaka, tahap selanjutnya adalah membuat analisis kebutuhan. Analisis kebutuhan ini penting karena akan menjadi acuan dalam mengimplementasikan kode program untuk menghindari kesalahan konsep perancangan algoritme. Analisis kebutuhan tersebut berupa fokus objek yang digunakan dalam penelitian, tempat pengambil objek penelitian, alat dan bahan penelitian, serta spesifikasi perangkat lunak yang digunakan untuk penelitian.

### 3.4 Implementasi

Proses implementasi dilakukan berdasarkan acuan dari rancangan algoritme serta analisis kebutuhan yang telah dibuat sebelumnya dengan menggunakan kode program yang ditulis dalam bahasa pemrograman Python. Pengimplementasian ini menggunakan bantuan *library* (numpy, pandas, cv2, dan os) sehingga dapat mempercepat proses komputasi program.

### 3.5 Pengujian dan Analisis

Proses pengujian dan analisis ini dilakukan setelah tahap implementasi. Pengujian dilakukan dengan tujuan untuk mengukur tingkat dari teknik pengenalan citra jenis makanan sehat dengan ekstraksi fitur *color channel* dan GLCM menggunakan metode K-NN ini. Terdapat tiga tahapan pengujian yang dilakukan, yaitu:

1. Pengujian nilai  $k$ .
2. Pengujian ekstraksi fitur.
3. Pengujian perhitungan jarak tetangga terdekat.

### 3.6 Kesimpulan

Pada tahap terakhir ini, penarikan kesimpulan dilakukan dengan menganalisa data terkait kesesuaian rumusan masalah dengan hasil pengujian data, serta hal-hal yang menjadikan suatu pengujian berhasil atau tidak, semuanya dituangkan di kesimpulan secara menyeluruh, padat, dan jelas. Sedangkan saran ditulis dengan tujuan untuk memberikan masukan untuk pengembangan penelitian.

### 3.7 Spesifikasi *Hardware* dan/atau *Software*

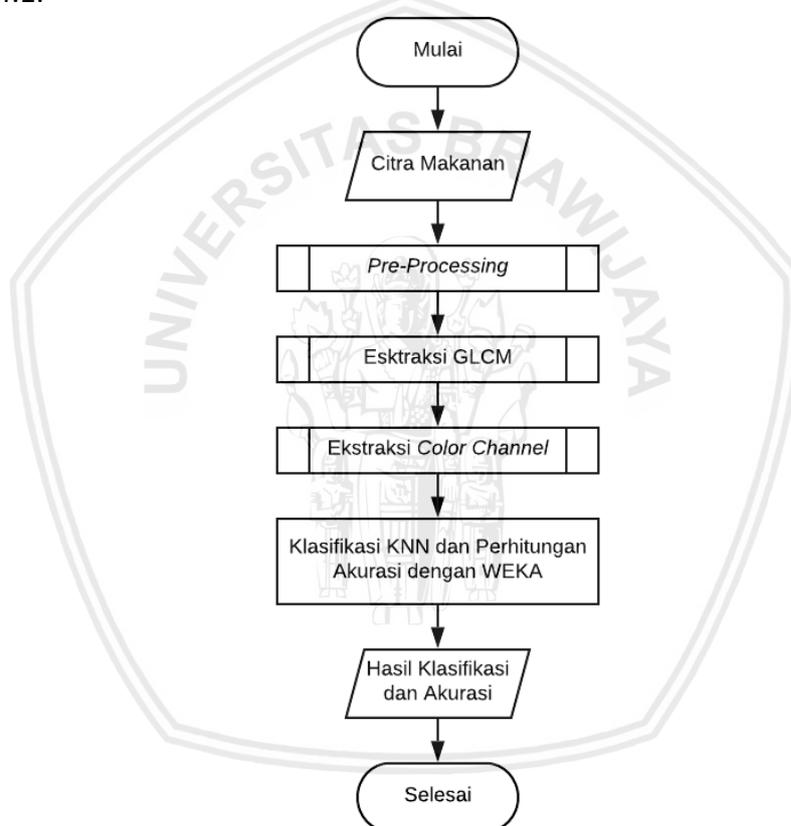
Untuk dapat menjalankan tahap pengujian dan analisis, berikut perangkat keras dan perangkat lunak yang digunakan:

- a. Perangkat keras : Laptop ASUS A456U 14 inches
  - Processor* : Intel Core I5-7200U, up to 3.18 Hz
  - RAM* : 4 GB
  - Harddisk* : 1 TB
- b. Perangkat lunak
  - Operating System* : Windows 10 Pro 64-bit (10.0, Build 10240)
  - Aplikasi pengujian* : Pycharm Edition 2018.2.4 dan WEKA 3.8

## BAB 4 PERANCANGAN

### 4.1 Perancangan Algoritme

Perancangan algoritme direpresentasikan dalam suatu diagram alir atau *flowchart* yang memberikan informasi tentang tahap-tahap pengerjaan program. Perancangan algoritme dibuat dengan tujuan agar alur suatu program dapat dengan mudah dipahami. Pada penelitian ini, alur pengerjaan program yang dilalui yaitu membaca masukan objek gambar makanan, *pre-processing* pada gambar, ekstraksi fitur tekstur dengan GLCM, ekstraksi fitur warna dengan *color channel*, klasifikasi dengan metode *K-Nearest Neighbor* (KNN), dan perhitungan akurasi. Perancangan algoritme tersebut ditunjukkan pada diagram berikut pada Gambar 4.1.



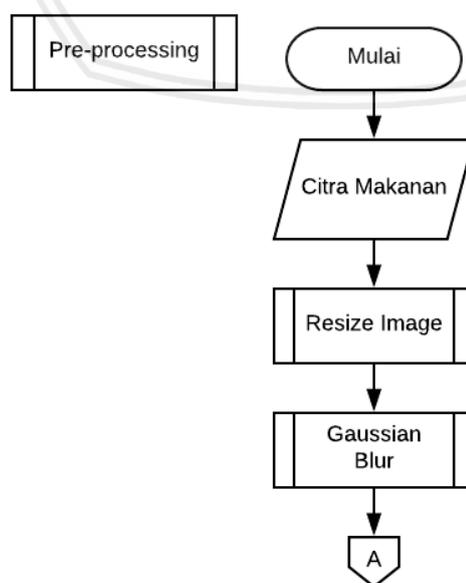
Gambar 4.1 Diagram Alir Sistem

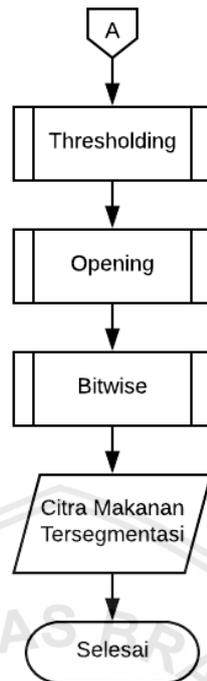
Berdasarkan diagram alir tersebut diatas dapat diketahui bahwa mulanya program akan membaca masukan citra berupa objek makanan. Selanjutnya akan dilakukan *pre-processing* untuk mendapatkan fokus objek yang akan digunakan untuk mengenali jenis makanan melalui proses segmentasi citra yang menghasilkan objek citra tersegmentasi. Hasil segmentasi citra tersebut kemudian digunakan untuk mendapatkan nilai tekstur menggunakan ekstraksi fitur tekstur GLCM. Jumlah fitur GLCM yang digunakan adalah sebanyak 8 fitur yang masing-masing memiliki 4 sudut. Selanjutnya hasil segmentasi citra juga dilakukan ekstraksi fitur warna menggunakan *color channel* untuk mendapatkan nilai warna pada objek yang akan dikenali jenis makanannya. Tipe warna yang

digunakan pada ekstraksi ini adalah tipe warna RGB (*Red, Green, Blue*) sedangkan fitur tekstur yang digunakan adalah *mean, skewness*, dan standar deviasi. Setelah berhasil mendapatkan nilai tekstur pada objek citra tersegmentasi, selanjutnya nilai-nilai tersebut akan dilakukan proses perhitungan kemiripan antara citra *query* dengan citra pada basis data dengan menggunakan *Euclidean distance*. Hasil pada perhitungan *Euclidean distance* pada masing-masing nilai fitur kemudian dilakukan klasifikasi menggunakan metode *K-Nearest Neighbor* (KNN) yang diurutkan berdasarkan nilai jarak terdekat antar citra *query* dengan citra pada basis data, sehingga jenis makanan dapat diketahui sebagai keluaran sistem.

#### 4.1.1 Pre-Processing

*Pre-processing* merupakan tahap dimana suatu citra masukan akan menghasilkan objek yang sudah tersegmentasi dengan menghilangkan *noise* yang ada pada citra masukan dengan cara mengubah ukuran citra terlebih dahulu. Selanjutnya citra tersebut dilakukan proses *filtering* menggunakan *gaussian filter* guna mereduksi derau yang bersifat normal serta banyak dijumpai pada sebaran citra hasil proses digitasi menggunakan sehingga citra akan terlihat lebih halus dan mudah untuk dilakukan klasifikasi. Selanjutnya dilakukan proses transformasi *channel* warna pada objek makanan menggunakan fungsi BGRTOHSV pada pemrograman Python. Objek makanan tersebut kemudian dilakukan *thresholding*. Fungsi dari *thresholding* adalah untuk mempertegas objek makanan yang sudah dideteksi. Objek makanan yang sudah dilakukan *thresholding* ini masih berupa citra HSV sehingga harus diubah terlebih dahulu menjadi citra biner. *Thresholding* ini berguna untuk menghilangkan background pada citra masukan secara otomatis. Proses terakhir yaitu *opening* dimana proses ini dapat memperhalus batas-batas objek, memisahkan objek yang sebelumnya bergandengan, serta menghilangkan objek yang memiliki ukuran lebih kecil daripada ukuran structuring. Diagram *pre-processing* ditunjukkan pada Gambar 4.2.

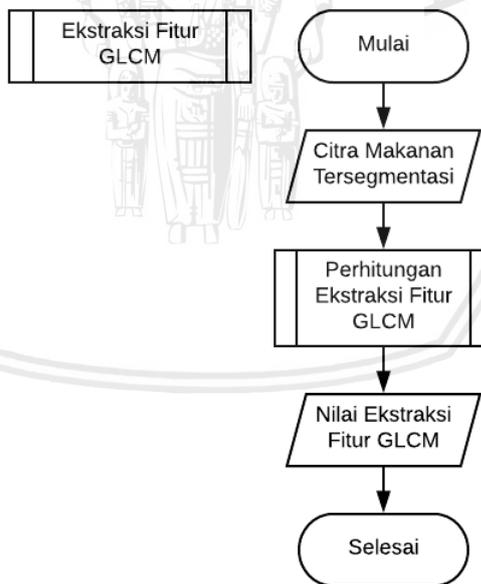




Gambar 4.2 Diagram Alir Pre-Processing

#### 4.1.2 Ekstraksi Tekstur GLCM

Tahapan ekstraksi tekstur GLCM ditunjukkan oleh diagram alir pada Gambar 4.3.



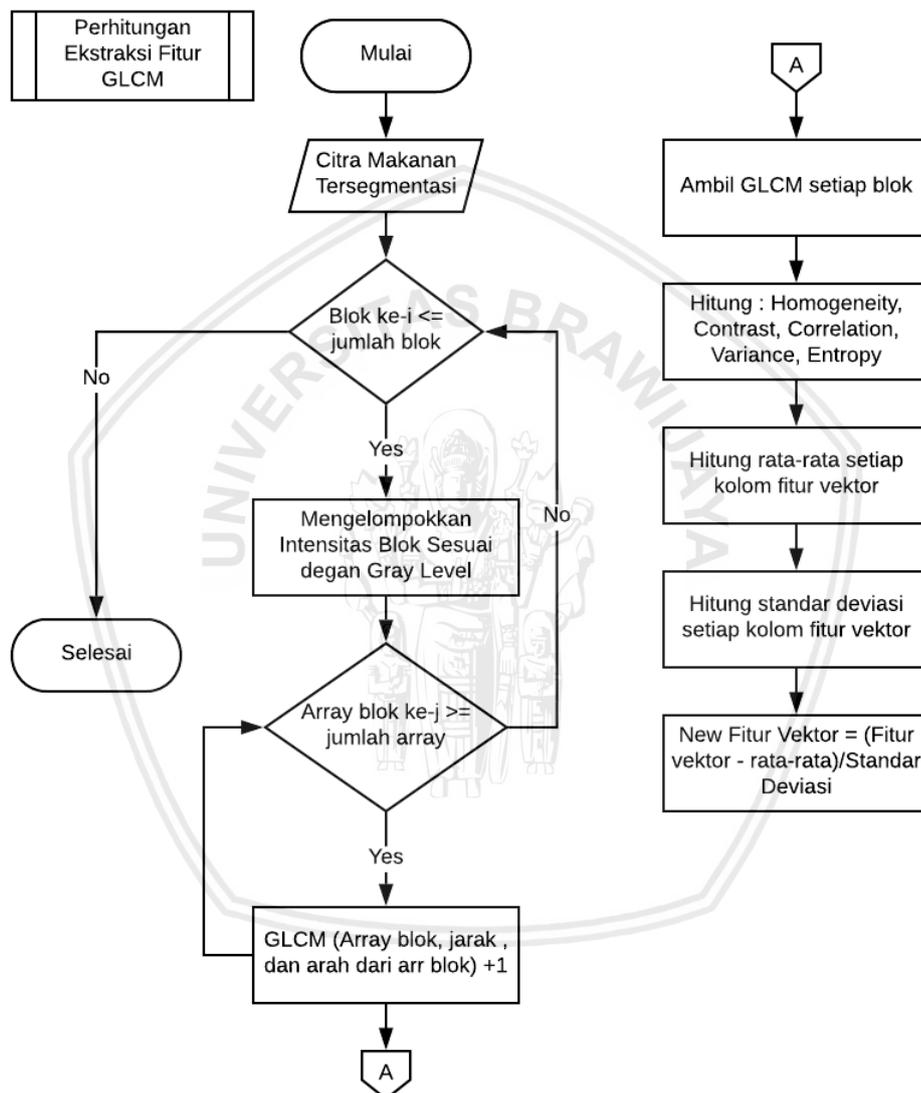
Gambar 4.3 Diagram Alir Ekstraksi Tesktur GLCM

Tahapan proses ekstraksi fitur GLCM ini menggunakan citra yang sudah dilakukan proses segmentasi. Prosesnya diawali dengan mencari nilai piksel terbesar dalam matriks citra tersegmentasi untuk mendapat ukuran matriks yang akan digunakan untuk menghitung jumlah semua nilai piksel dalam elemen matriks berdasarkan koordinat sumbu x dan y. Selanjutnya, nilai piksel di masing-masing elemen matriks tersebut dilakukan normalisasi dengan cara membagi

nilai piksel dengan total keseluruhan nilai yang ada pada matriks. Nilai matriks yang sudah ternormalisasi inilah yang digunakan untuk menghitung nilai ekstraksi fitur sebanyak 5 fitur yaitu kontras, homogenitas, entropi, variansi, dan korelasi.

#### 4.1.2.1 Perhitungan Fitur GLCM

Diagram alir perhitungan ekstraksi fitur tesktur dengan metode GLCM ini ditunjukkan pada Gambar 4.4 berikut.



**Gambar 4.4 Diagram alir perhitungan matriks GLCM**

Penjelasan terkait alur perhitungan matriks GLCM adalah sebagai berikut:

1. Citra tersegmentasi diterima sebagai masukan pada fungsi dalam program.
2. Program membaca ukuran citra tersegmentasi, kemudian membuat urutan baru dengan kernel berukuran 256x256 untuk menyimpan hasil perhitungan.
3. Program melakukan perhitungan matriks awal GLCM yaitu menggunakan 4 arah sudut berupa sudut 0°, 45°, 90°, dan 135°. Ketentuan setiap sudutnya adalah apabila sudut 0°, perhitungan matriks awal GLCM dilakukan dengan

melihat banyaknya kemunculan nilai piksel dengan nilai piksel tetangga secara linier kanan atau kiri yang jaraknya 1 piksel. Apabila sudut  $45^\circ$ , perhitungan matriks awal GLCM dilakukan dengan melihat banyaknya kemunculan nilai piksel dengan nilai piksel tetangga secara diagonal kanan ke atas atau ke bawah yang jaraknya 1 piksel. Apabila sudut  $90^\circ$ , perhitungan matriks awal GLCM dilakukan dengan melihat banyaknya kemunculan nilai piksel dengan nilai piksel tetangga secara linier ke atas atau ke bawah yang jaraknya 1 piksel. Apabila sudut  $135^\circ$ , perhitungan matriks awal GLCM dilakukan dengan melihat banyaknya kemunculan nilai piksel dengan nilai piksel tetangga secara diagonal ke kanan atas atau ke bawah yang jaraknya 1 piksel.

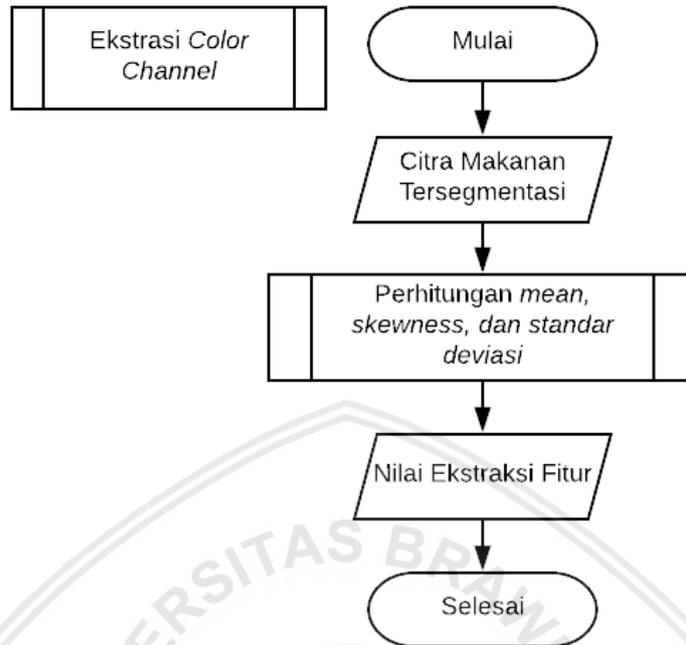
4. Pembentukan matriks secara simetris melalui operasi penjumlahan pada matriks awal GLCM dengan transposenya.
5. Melakukan normalisasi matriks melalui operasi pembagian dengan matriks total nilai piksel pada citra tersegmentasi.
6. Program menghasilkan keluaran berupa matriks GLCM yang sudah ternormalisasi.

Berdasarkan diagram alir perhitungan ekstraksi fitur tekstur dengan metode GLCM ini ditunjukkan pada Gambar 4.4 tersebut dapat kita lihat bahwa perhitungan GLCM dihasilkan dengan mengetahui tingkat keabuan yang sering muncul dalam citra makanan yang tersegmentasi. Metode ini juga dapat disebut sebagai metode perhitungan yang mengkalkulasi frekuensi kombinasi nilai piksel yang sering muncul pada citra tersegmentasi. Dari hasil perhitungan kombinasi nilai piksel ini lah didapatkan nilai-nilai fitur ekstraksi yang digunakan sebagai alat untuk melakukan analisis lanjutan berdasarkan statistik sebaran nilai piksel dari intensitas pikselnya. Dari piksel-piksel tersebutlah akan terjadi pembentukan matrik ko-okurensi dengan pasangan pikselnya. Keberadaan matriks ini dapat dilihat pada suatu kondisi dimana nilai piksel akan melakukan perulangan sehingga menemukan pasangan piksel keabuannya. Kondisi ini disimbolkan dengan matriks yang memiliki dua posisi jarak yaitu  $(x_1, y_1)$  serta  $(x_2, y_2)$ . Dua kondisi tersebut menunjukkan bahwa dibutuhkan suatu hal lain yang dapat membedakan nilai fitur satu citra tersegmentasi dengan citra tersegmentasi lainnya yang disebut sebagai fitur ekstraksi.

#### 4.1.3 Ekstraksi Warna dengan *Color Channel*

Pada proses ekstraksi warna dengan *color channel* ini terlebih dahulu dilakukan pembacaan data berupa citra makanan tersegmentasi. Citra makanan tersegmentasi tersebut kemudian dilakukan perhitungan fitur ekstraksi warna yang meliputi 3 jenis yaitu mean, *skewness*, dan standar deviasi. Fitur *mean* berfungsi untuk mendapatkan rata-rata nilai piksel pada masing-masing *channel* yaitu R, G, dan B berdasarkan ukuran pikselnya. Fitur *skewness* memiliki rumus atau formula yang sama yaitu mendapatkan nilai rata-rata piksel dikurangi dengan hasil perhitungan pada fitur *mean*. Keduanya memiliki perbedaan pada nilai  $n$  nya berbeda yang masing-masing bernilai 3 dan 4. Sedangkan standar deviasi berfungsi untuk mengetahui sebaran nilai piksel pada citra makanan tersegmentasi.

Tahapan ekstraksi warna *Color Channel* ditunjukkan oleh diagram alir pada Gambar 4.6.

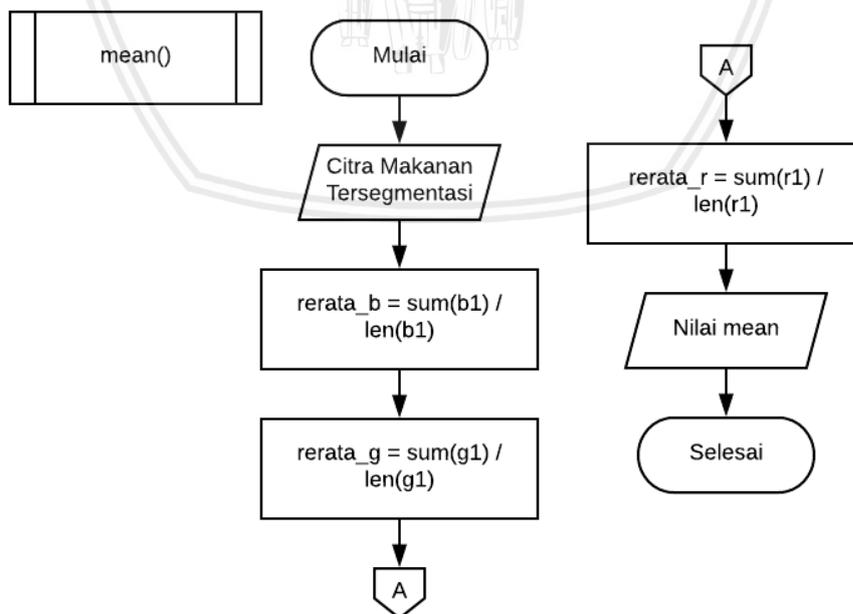


**Gambar 4.5 Diagram Alir Ekstraksi *Color Channel***

Hasil perhitungan fitur-fitur tersebut inilah yang nantinya akan digunakan sebagai nilai yang mengidentifikasi kelas makanan pada citra kueri.

#### 4.1.3.1 Perhitungan Ekstraksi Fitur *Color Channel*

Diagram alir perhitungan ekstraksi fitur *mean* dengan metode *color channel* ini ditunjukkan pada Gambar 4.7 berikut.

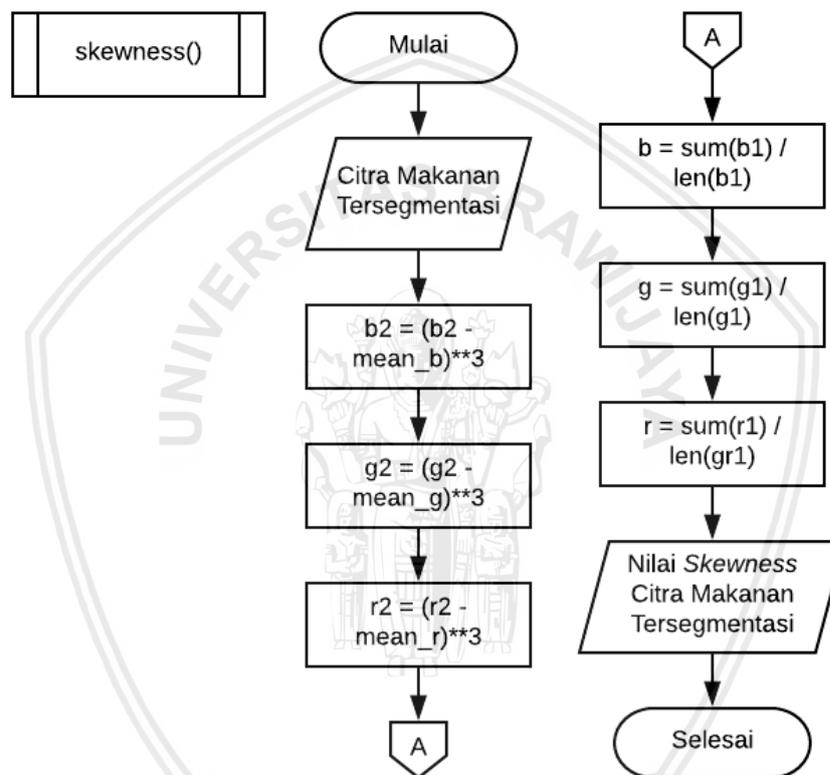


**Gambar 4.6 Diagram alir perhitungan rata-rata pada citra tersegmentasi**

Penjelasan terkait alur perhitungan ekstraksi fitur *mean* adalah sebagai berikut:

1. Program membaca masukan berupa citra makanan tersegmentasi.
2. Program melakukan perhitungan ekstraksi fitur *color channel* dengan terlebih dahulu membuat array untuk menyimpan nilai ekstraksi pada tiap ruang warna untuk *Red, Green, dan Blue*.
3. Perhitungan rata-rata pada tiap ruang warna untuk *Red, Green, dan Blue* yang disimpan dalam variabel *rerata\_b, rerata\_g, dan rerata\_r*.
4. Hasil keluran berupa nilai ekstraksi warna masing-masing pada ruang warna RGB yaitu *Red, Green, dan Blue*.

Selanjutnya, diagram alir perhitungan nilai standar deviasi pada citra tersegmentasi ditunjukkan pada Gambar 4.8 berikut.



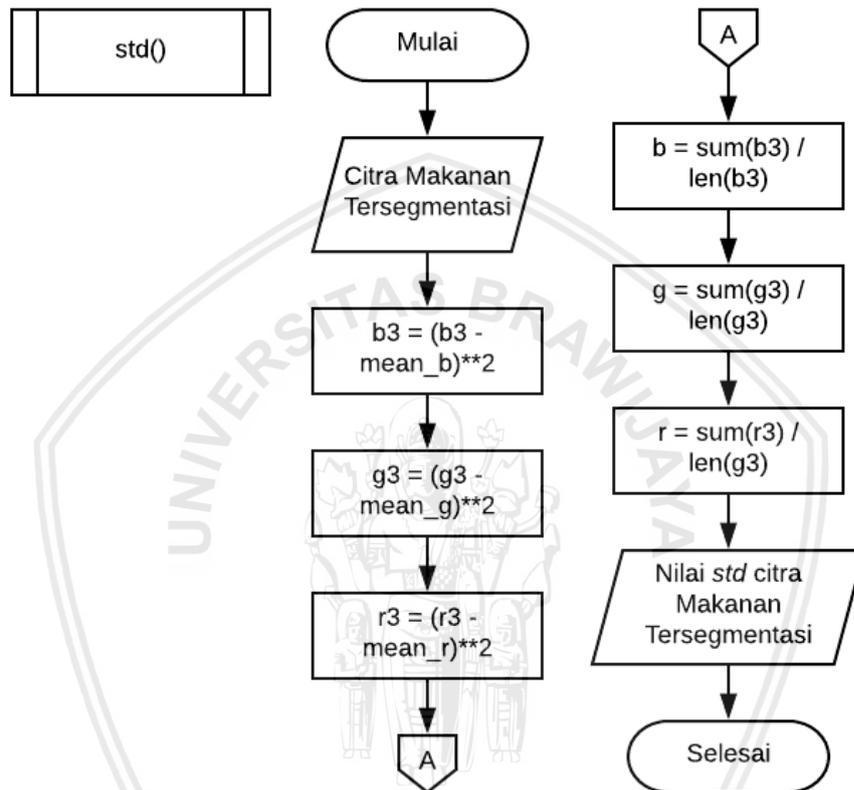
**Gambar 4.7 Diagram alir perhitungan *skewness* pada citra tersegmentasi**

Penjelasan terkait alur perhitungan ekstraksi fitur *skewness* adalah sebagai berikut:

1. Program membaca masukan berupa nilai ekstraksi fitur *mean*.
2. Program melakukan perhitungan ekstraksi fitur *color channel* dengan terlebih dahulu membuat array untuk menyimpan nilai ekstraksi pada tiap ruang warna untuk *Red, Green, dan Blue* pada variabel *skew\_b, skew\_g, dan skew\_r*.
3. Program mengambil data berupa nilai ekstraksi fitur *mean* disimpan dalam variabel *mean\_b, mean\_g, dan mean\_r*.
4. Perhitungan pada tiap ruang warna untuk *Red, Green, dan Blue* melalui operasi pangkat tiga terhadap hasil operasi pengurangan nilai piksel pada variabel *b2, g2, dan r2* dengan nilai hasil ekstraksi fitur *mean\_b, mean\_g, dan mean\_r*.

- Perhitungan pada tiap ruang warna untuk *Red*, *Green*, dan *Blue* melalui operasi pembagian nilai total piksel pada variabel  $b_2$ ,  $g_2$ , dan  $r_2$  dengan ukuran piksel  $b_2$ ,  $g_2$ , dan  $r_2$ .
- Hasil keluran berupa nilai ekstraksi warna masing-masing pada ruang warna RGB yaitu *Red*, *Green*, dan *Blue* yang disimpan pada variabel variabel  $skew\_b$ ,  $skew\_g$ , dan  $skew\_r$ .

Selanjutnya, diagram alir perhitungan nilai standar deviasi pada citra tersegmentasi ditunjukkan pada Gambar 4.8 berikut.



**Gambar 4.8** Diagram alir perhitungan standar deviasi pada citra tersegmentasi

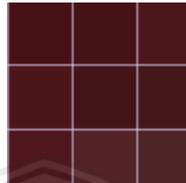
Penjelasan terkait alur perhitungan ekstraksi fitur *standar deviasi* adalah sebagai berikut:

- Program membaca masukan berupa nilai ekstraksi fitur *skewness*.
- Program melakukan perhitungan ekstraksi fitur *color channel* dengan terlebih dahulu membuat array untuk menyimpan nilai ekstraksi pada tiap ruang warna untuk *Red*, *Green*, dan *Blue* pada variabel  $std\_b$ ,  $std\_g$ , dan  $std\_r$ .
- Program mengambil data berupa nilai ekstraksi fitur mean disimpan dalam variabel  $skew\_b$ ,  $skew\_g$ , dan  $skew\_r$ .
- Perhitungan pada tiap ruang warna untuk *Red*, *Green*, dan *Blue* melalui operasi pangkat dua terhadap hasil operasi pengurangan nilai piksel pada variabel  $b_3$ ,  $g_3$ , dan  $r_3$  dengan nilai hasil ekstraksi fitur  $skew\_b$ ,  $skew\_g$ , dan  $skew\_r$ .
- Perhitungan pada tiap ruang warna untuk *Red*, *Green*, dan *Blue* melalui operasi pembagian nilai total piksel pada variabel  $b_3$ ,  $g_3$ , dan  $r_3$  dengan ukuran piksel  $b_3$ ,  $g_3$ , dan  $r_3$ .

6. Hasil keluran berupa nilai ekstraksi warna masing-masing pada ruang warna RGB yaitu *Red*, *Green*, dan *Blue* yang disimpan pada variabel variabel *std\_b*, *std\_g*, dan *std\_r*.

## 4.2 Perhitungan Manual

Penerapan perhitungan manual pada citra makanan tersegmentasi dilakukan dengan mengambil sampel citra berukuran 3x3 piksel terhadap citra asli. Pada Gambar 4.15 ditunjukkan sampel citra makanan tersegmentasi.



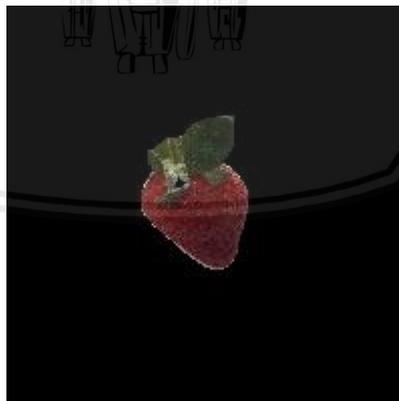
**Gambar 4.9** Sampel citra makanan berukuran 3x3 piksel

Berdasarkan sampel citra tersebut di atas, pada Gambar 4.16 menunjukkan nilai *greyscale* pada setiap piksel.

Greyscale			
x,y	0	1	2
0	35	30	35
1	34	38	33
2	35	37	32

**Gambar 4.10** Nilai piksel *greyscale* pada sampel citra tersegmentasi

Citra asli tersegmentasi berukuran 200x200 yang merupakan hasil dari *preprocessing* ditunjukkan pada Gambar 4.17.



**Gambar 4.11** Citra asli tersegmentasi berukuran 200x200

### 4.2.1 Ekstraksi Tekstur GLCM

Citra hasil *pre-processing* pada tahap sebelumnya kemudian digunakan pada proses berikutnya yaitu proses ekstraksi fitur tesktur menggunakan metode GLCM. Berikut dijabarkan tahap-tahap perhitungan metode GLCM untuk dapat mengekstraksi nilai tekstur pada citra hasil *pre-processing*.

- Perhitungan matriks *co-occurrence* menggunakan 4 sudut yaitu  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$  serta pengaturan jarak antar piksel yaitu 1 piksel ( $d = 1$ ). Gambar 4.18 menunjukkan hasil pembentukan matrik ko-okurensi pada piksel (0,0).

a) Sudut arah  $0^\circ$

35	30	35
34	38	33
35	37	32

**Gambar 4.12** Gambar matriks *co-occurrence* piksel (35,30) dan  $d = 1$  sudut  $0^\circ$

Dapat diketahui pada Gambar 4.18 tersebut di atas bahwa matriks dengan sudut arah  $0^\circ$  memiliki 1 pasang ko-okurensi pada piksel (35,30). Perhitungan jumlah ko-okurensi piksel tersebut dilakukan dengan menghitung intensitas banyaknya nilai pasangan piksel kemudian diletakkan pada tabel ko-okurensi seperti pada Gambar 4.19 berikut yang menunjukkan nilai matriks pada sudut  $0^\circ$  dimana nilai  $x$  merupakan piksel pada sumbu ke kanan sedangkan  $y$  merupakan piksel pada sumbu ke bawah.

x,y	0	...	30	33	34	35	38
0	0	...	0	0	0	0	0
...	...	...	...	...	...	...	...
30	0	...	0	0	0	1	0
33	0	...	0	0	0	0	1
34	0	...	0	0	0	0	0
35	0	...	1	0	0	0	0
38	0	...	0	0	1	0	0

**Gambar 4.13** Tabel *co-occurrence* GLCM untuk Sudut  $0^\circ$  dari Gambar 4.18

b) Sudut arah  $45^\circ$

35	30	35
34	38	33
35	37	32

**Gambar 4.14** Gambar matriks *co-occurrence* piksel (34,30) dan  $d = 1$  sudut  $45^\circ$

Dapat diketahui pada Gambar 4.20 tersebut di atas bahwa matriks dengan sudut arah  $45^\circ$  memiliki 1 pasang ko-okurensi pada piksel (34,30). Perhitungan jumlah ko-okurensi piksel tersebut dilakukan dengan menghitung intensitas banyaknya nilai pasangan piksel kemudian diletakkan pada tabel ko-okurensi seperti pada Gambar 4.21 berikut yang menunjukkan nilai matriks pada sudut  $45^\circ$  dimana nilai  $x$  merupakan piksel pada sumbu ke kanan sedangkan  $y$  merupakan piksel pada sumbu ke bawah.



x,y	0	...	30	33	34	35	38
0	0	...	0	0	0	0	0
...	...	...	...	...	...	...	...
30	0	...	0	0	0	0	0
33	0	...	0	0	0	0	0
34	0	...	1	0	0	0	0
35	0	...	0	0	0	0	1
38	0	...	0	0	0	1	0

Gambar 4.15 Tabel co-occurrence GLCM untuk Sudut 45° dari Gambar 4.20

c) Sudut arah 90°

35	30	35
34	38	33
35	37	32

Gambar 4.16 Gambar matriks co-occurrence piksel (35,34) dan d = 1 sudut 90°

Dapat diketahui pada Gambar 4.22 tersebut di atas bahwa matriks dengan sudut arah 90° memiliki 1 pasang ko-okurensi pada piksel (35,34). Perhitungan jumlah ko-okurensi piksel tersebut dilakukan dengan menghitung intensitas banyaknya nilai pasangan piksel kemudian diletakkan pada tabel ko-okurensi seperti pada Gambar 4.23 berikut yang menunjukkan nilai matriks pada sudut 90° dimana nilai x merupakan piksel pada sumbu ke kanan sedangkan y merupakan piksel pada sumbu ke bawah.

x,y	0	...	30	33	34	35	38
0	0	...	0	0	0	0	0
...	...	...	...	...	...	...	...
30	0	...	0	0	0	0	1
33	0	...	0	0	0	0	0
34	0	...	0	0	0	1	0
35	0	...	0	1	1	0	1
38	0	...	0	0	0	0	0

Gambar 4.17 Tabel co-occurrence GLCM untuk Sudut 90° dari Gambar 4.22

d) Sudut arah 135°

35	30	35
34	38	33
35	37	32

Gambar 4.18 Gambar matriks co-occurrence piksel (30,33) dan d = 1 sudut 135°

Dapat diketahui pada Gambar 4.24 tersebut di atas bahwa matriks dengan sudut arah 135° memiliki 1 pasang ko-okurensi pada piksel (30,33). Perhitungan jumlah ko-okurensi piksel tersebut dilakukan dengan menghitung intensitas banyaknya nilai pasangan piksel kemudian diletakkan pada tabel ko-okurensi seperti pada Gambar 4.25 berikut yang menunjukkan nilai matriks pada sudut 135° dimana

nilai x merupakan piksel pada sumbu ke kanan sedangkan y merupakan piksel pada sumbu ke bawah.

x,y	0	...	30	33	34	35	38
0	0	...	0	0	0	0	0
...	...	...	...	...	...	...	...
30	0	...	0	1	0	0	0
33	0	...	0	0	0	0	0
34	0	...	0	0	0	0	0
35	0	...	0	0	0	0	1
38	0	...	0	0	0	0	0

Gambar 4.19 Tabel co-occurrence GLCM untuk Sudut 135° dari Gambar 4.24

2. Pembentukan matriks *co-occurrence* secara simetris dengan melakukan operasi penjumlahan antara matriks GLCM pada perhitungan di poin 1 dan matriks transposenya.

a) Sudut arah 0°

$$\begin{aligned}
 GLCM &= \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 1 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 1 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 &= \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 2 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 2 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
 \end{aligned}$$

b) Sudut arah 45°

$$\begin{aligned}
 GLCM &= \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 1 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 1 \\ 0 & \dots & 0 & 0 & 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 1 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 1 \\ 0 & \dots & 0 & 0 & 0 & 1 & 0 \end{bmatrix}
 \end{aligned}$$



$$= \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 1 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 1 & 0 & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 2 \\ 0 & \dots & 0 & 0 & 0 & 2 & 0 \end{bmatrix}$$

c) Sudut arah 90°

$$GLCM = \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & 1 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 1 & 0 \\ 0 & \dots & 0 & 1 & 1 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & 1 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 1 & 0 \\ 0 & \dots & 0 & 1 & 1 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & 1 \\ 0 & \dots & 0 & 0 & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & 0 & 2 & 0 \\ 0 & \dots & 0 & 1 & 2 & 0 & 0 \\ 0 & \dots & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

d) Sudut arah 135°

$$GLCM = \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 1 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 1 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 0 & 0 & 0 \\ 0 & \dots & 1 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 1 & 0 & 0 & 0 & 1 \\ 0 & \dots & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

3. Langkah selanjutnya adalah melakukan normalisasi pada matriks *co-occurrence* yang berfungsi untuk mencari nilai kemungkinan pada tiap pasangan piksel. Prosesnya dilakukan dengan cara membagi antar nilai piksel pada dengan jumlah total nilai piksel setiap besaran sudut.

a) Sudut 0°

Untuk besaran sudut  $0^\circ$  jumlah total nilai piksel pada matriks *co-occurrence* nya adalah 6 piksel. Perhitungan berikut menunjukkan langkah-langkah mendapatkan probabilitas nilai piksel pada matriks *co-occurrence*.

$$GLCM = \frac{\begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 2 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 2 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}{6}$$

$$= \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0,33 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0,33 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

b) Sudut  $45^\circ$

Untuk besaran sudut  $45^\circ$  jumlah total nilai piksel pada matriks *co-occurrence* nya adalah 4 piksel. Perhitungan berikut menunjukkan langkah-langkah mendapatkan probabilitas nilai piksel pada matriks *co-occurrence*.

$$GLCM = \frac{\begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 1 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 1 & 0 & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 2 \\ 0 & \dots & 0 & 0 & 0 & 2 & 0 \end{bmatrix}}{4}$$

$$= \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0,25 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0,25 & 0 & 0 & 0,25 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0,5 \\ 0 & \dots & 0 & 0 & 0 & 0,5 & 0 \end{bmatrix}$$

c) Sudut  $90^\circ$

Untuk besaran sudut  $90^\circ$  jumlah total nilai piksel pada matriks *co-occurrence* nya adalah 6 piksel. Perhitungan berikut menunjukkan langkah-langkah mendapatkan probabilitas nilai piksel pada matriks *co-occurrence*.



$$GLCM = \frac{\begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & 1 \\ 0 & \dots & 0 & 0 & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & 0 & 2 & 0 \\ 0 & \dots & 0 & 1 & 2 & 0 & 0 \\ 0 & \dots & 1 & 0 & 0 & 0 & 0 \end{bmatrix}}{6}$$

$$= \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & 0,17 \\ 0 & \dots & 0 & 0 & 0 & 0,17 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0,33 & 0 \\ 0 & \dots & 0 & 0,17 & 0,33 & 0 & 0 \\ 0 & \dots & 0,17 & 0 & 0 & 0 & 0 \end{bmatrix}$$

d) Sudut 135°

Untuk besaran sudut 135° jumlah total nilai piksel pada matriks *co-occurrence* nya adalah 4 piksel. Perhitungan berikut menunjukkan langkah-langkah mendapatkan probabilitas nilai piksel pada matriks *co-occurrence*.

$$GLCM = \frac{\begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 0 & 0 & 0 \\ 0 & \dots & 1 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 1 & 0 & 0 & 0 & 1 \\ 0 & \dots & 0 & 0 & 0 & 1 & 0 \end{bmatrix}}{4}$$

$$= \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0,25 & 0 & 0 & 0 \\ 0 & \dots & 0,25 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0,25 & 0 & 0 & 0 & 0,25 \\ 0 & \dots & 0 & 0 & 0 & 0,25 & 0 \end{bmatrix}$$

4. Menghitung nilai ekstraksi fitur menggunakan nilai piksel yang sudah dinormalisasi pada masing-masing sudut. Pada penelitian ini terdapat 5 jenis fitur GLCM yang digunakan seperti *homogeneity*, *contrast*, *correlation*, *variance* dan *entropy*. Berikut ditunjukkan contoh perhitungan kelima fitur tersebut.

a) Sudut 0°

$$\begin{aligned} Homogeneity &= 0^2 + 0^2 + 0,33^2 + 0,33^2 + \dots + 0,33^2 \\ &= 0^2 + 0^2 + 0,1089 + 0,1089 + \dots + 0,1089 \\ &= 0,6534 \end{aligned}$$

$$\begin{aligned} Contrast &= ((0 - 0)^2 * 0) + ((0 - 0)^2 * 0) + ((35 - 30)^2 * 0,33) + \\ & \quad ((38 - 34)^2 * 0,33) + \dots + ((37 - 32)^2 * 0,33) \end{aligned}$$



$$= 0 + 0 + 16,5 + 10,56 + \dots + 16,5$$

$$= 39,6$$

Perhitungan correlation dilakukan dengan terlebih dahulu menghitung rerata nilai  $P_x$  dan  $P_y$ , serta standar deviasi nilai  $P_x$  dan  $P_y$ .

$$\mu_x = (0 * 0) + (0 * 0) + (35 * 0,33) + (38 * 0,33) + \dots + (37 * 0,33)$$

$$= (0) + (0) + (11,55) + (12,54) + \dots + (12,21)$$

$$= 69,25$$

$$\mu_y = (0 * 0) + (0 * 0) + (30 * 0,33) + (33 * 0,33) + \dots + (32 * 0,33)$$

$$= (0) + (0) + (9,9) + (10,89) + \dots + (10,56)$$

$$= 59,02$$

$$\sigma_x = (0 * \sqrt{((0 - 69,25)^2)}) + (0 * \sqrt{((0 - 69,25)^2)}) + (0,33 * \sqrt{((35 - 69,25)^2)}) + (0,33 * \sqrt{((38 - 69,25)^2)}) + \dots + (0,33 * \sqrt{((37 - 69,25)^2)})$$

$$= (0) + (0) + (11,3) + (10,31) + \dots + (10,64)$$

$$= 68,145$$

$$\sigma_y = (0 * \sqrt{((0 - 69,25)^2)}) + (0 * \sqrt{((0 - 69,25)^2)}) + (0,33 * \sqrt{((30 - 69,25)^2)}) + (0,33 * \sqrt{((33 - 69,25)^2)}) + \dots + (0,33 * \sqrt{((32 - 69,25)^2)})$$

$$= (0) + (0) + (9,57) + (8,58) + \dots + (8,91)$$

$$= 49,209$$

$$Correlation = \left( \frac{(((0 - 0) * 0) - (69,25 - 59,02))}{68,145 * 49,209} \right) + \left( \frac{(((38 - 34) * 0,33) - (69,25 - 59,02))}{68,145 * 49,209} \right) + \dots + \left( \frac{(((37 - 32) * 0,33) - (69,25 - 59,02))}{68,145 * 49,209} \right)$$

$$= (0) + (-0,0034) + \dots + (-0,0026)$$

$$= -0,0179$$

Selanjutnya, menghitung nilai variansi yang dilakukan dengan terlebih dahulu menghitung nilai rerata P.

$$\mu = \frac{(0 + 0 + 0,33 + 0,33 + \dots + 0,33)}{1444}$$

$$= \frac{1,98}{1444}$$

$$= 0,00137$$

$$Variance = ((0 - 0,00137)^2 * 0) + ((30 - 0,00137)^2 * 0,33) + \dots + ((32 - 0,00137)^2 * 0,33)$$

$$= 0 + 296,9729 + \dots + 337,8911$$

$$= 3178,64$$

$$Entropy = -((0) * \log(0,33)) + (-(0,33) * \log(0,33)) + \dots + (-(0,33) * \log(0,33))$$

$$= 0 + 0,1584 + \dots + 0,1584$$



$$= 0,9584$$

b) Sudut 45°

$$\begin{aligned} \text{Homogeneity} &= 0^2 + 0^2 + 0,25^2 + 0,25^2 + \dots + 0,5^2 \\ &= 0^2 + 0^2 + 0,0625 + 0,0625 + \dots + 0,25 \\ &= 0,6875 \end{aligned}$$

$$\begin{aligned} \text{Contrast} &= ((0 - 0)^2 * 0) + ((0 - 0)^2 * 0) + ((38 - 35)^2 * 0,5) + \\ &\quad ((34 - 30)^2 * 0,25) + \dots + ((37 - 33)^2 * 0,25) \\ &= 0 + 0 + 4 + 4,5 + \dots + 4,5 \\ &= 17 \end{aligned}$$

Perhitungan correlation dilakukan dengan terlebih dahulu menghitung rerata nilai  $P_x$  dan  $P_y$ , serta standar deviasi nilai  $P_x$  dan  $P_y$ .

$$\begin{aligned} \mu_x &= (0 * 0) + (0 * 0) + (35 * 0,5) + (38 * 0,5) + \dots + (37 * 0,25) \\ &= (0) + (0) + (17,5) + (19) + \dots + (18,5) \\ &= 72 \end{aligned}$$

$$\begin{aligned} \mu_y &= (0 * 0) + (0 * 0) + (30 * 0,25) + (33 * 0,25) + \dots + (38 * 0,5) \\ &= (0) + (0) + (7,5) + (8,25) + \dots + (19) \\ &= 52,25 \end{aligned}$$

$$\begin{aligned} \sigma_x &= \left(0 * \sqrt{((0 - 72)^2)}\right) + \left(0 * \sqrt{((0 - 72)^2)}\right) + \left(0,5 * \sqrt{((35 - 72)^2)}\right) + \\ &\quad \left(0,25 * \sqrt{((38 - 69,25)^2)}\right) + \dots + \left(0,25 * \sqrt{((37 - 69,25)^2)}\right) \\ &= (0) + (0) + (9,25) + (17) + \dots + (8,75) \\ &= 73,5 \end{aligned}$$

$$\begin{aligned} \sigma_y &= \left(0 * \sqrt{((0 - 52,25)^2)}\right) + \left(0 * \sqrt{((0 - 52,25)^2)}\right) + \left(0,25 * \sqrt{((30 - 52,25)^2)}\right) + \\ &\quad \left(0,25 * \sqrt{((33 - 52,25)^2)}\right) + \dots + \left(0,25 * \sqrt{((32 - 52,25)^2)}\right) \\ &= (0) + (0) + (5,5625) + (4,8125) + \dots + (5,0625) \\ &= 35 \end{aligned}$$

$$\begin{aligned} \text{Correlation} &= \left(\frac{(((0 - 0) * 0) - (72 - 52,25))}{73,5 * 35}\right) \\ &\quad + \left(\frac{(((34 - 30) * 0,25) - (72 - 52,25))}{73,5 * 35}\right) + \dots \\ &\quad + \left(\frac{(((37 - 33) * 0,25) - (72 - 52,25))}{73,5 * 35}\right) \\ &= (0) + (-0,00729) + \dots + (-0,00729) \\ &= -0,02993 \end{aligned}$$

Selanjutnya, menghitung nilai variansi yang dilakukan dengan terlebih dahulu menghitung nilai rerata P.

$$\begin{aligned} \mu &= \frac{(0 + 0 + 0,25 + 0,25 + \dots + 0,5)}{1444} \\ &= \frac{1,75}{1444} \\ &= 0,00121 \end{aligned}$$



$$\begin{aligned} \text{Variance} &= ((0 - 0,00121)^2 * 0) + ((30 - 0,00121)^2 * 0,25) + \dots + ((32 - 0,00121)^2 * 0,25) \\ &= 0 + 224,9819 + \dots + 255,9806 \\ &= 3075,288 \end{aligned}$$

$$\begin{aligned} \text{Entropy} &= -((0,25) * \log(0,25)) + \dots + (-((0,25) * \log(0,25))) \\ &= 0 + 0,1505 + \dots + 0,1505 \\ &= 0,6004 \end{aligned}$$

c) Sudut 90°

$$\begin{aligned} \text{Homogeneity} &= 0^2 + 0^2 + 0,17^2 + 0,17^2 + \dots + 0,33^2 \\ &= 0^2 + 0^2 + 0,00289 + 0,00289 + \dots + 0,1089 \\ &= 0,12335 \end{aligned}$$

$$\begin{aligned} \text{Contrast} &= ((0 - 0)^2 * 0) + ((0 - 0)^2 * 0) + ((35 - 34)^2 * 0,17) + \\ &\quad ((30 - 38)^2 * 0,17) + \dots + ((33 - 32)^2 * 0,17) \\ &= 0 + 0 + 0,17 + 10,88 + \dots + 0,17 \\ &= 12,88 \end{aligned}$$

Perhitungan correlation dilakukan dengan terlebih dahulu menghitung rerata nilai  $P_x$  dan  $P_y$ , serta standar deviasi nilai  $P_x$  dan  $P_y$ .

$$\begin{aligned} \mu_x &= (0 * 0) + (0 * 0) + (35 * 0,33) + (38 * 0,17) + \dots + (34 * 0,17) \\ &= (0) + (0) + (11,55) + (6,46) + \dots + (5,78) \\ &= 51,33 \end{aligned}$$

$$\begin{aligned} \mu_y &= (0 * 0) + (0 * 0) + (34 * 0,17) + (38 * 0,17) + \dots + (33 * 0,33) \\ &= (0) + (0) + (5,78) + (6,46) + \dots + (10,89) \\ &= 57,45 \end{aligned}$$

$$\begin{aligned} \sigma_x &= (0 * \sqrt{((0 - 51,33)^2)}) + (0 * \sqrt{((0 - 51,33)^2)}) + (0,33 * \\ &\quad \sqrt{((35 - 51,33)^2)}) + (0,17 * \sqrt{((38 - 51,33)^2)}) + \dots + (0,17 * \\ &\quad \sqrt{((34 - 51,33)^2)}) \\ &= (0) + (0) + (5,3889) + (2,2661) + \dots + (2,9461) \\ &= 25,665 \end{aligned}$$

$$\begin{aligned} \sigma_y &= (0 * \sqrt{((0 - 57,45)^2)}) + (0 * \sqrt{((0 - 57,45)^2)}) + (0,17 * \\ &\quad \sqrt{((34 - 57,45)^2)}) + (0,33 * \sqrt{((33 - 57,45)^2)}) + \dots + (0,17 * \\ &\quad \sqrt{((32 - 57,45)^2)}) \\ &= (0) + (0) + (3,9865) + (8,0685) + \dots + (4,3265) \\ &= 30,573 \end{aligned}$$

$$\begin{aligned} \text{Correlation} &= \left( \frac{(((0 - 0) * 0) - (51,33 - 57,45))}{25,665 * 30,573} \right) \\ &\quad + \left( \frac{(((35 - 34) * 0,25) - (51,33 - 57,45))}{25,665 * 30,573} \right) + \dots \\ &\quad + \left( \frac{(((30 - 38) * 0,25) - (51,33 - 57,45))}{25,665 * 30,573} \right) \\ &= (0) + (0,008016) + \dots + (0,006066) \\ &= 0,046339 \end{aligned}$$



Selanjutnya, menghitung nilai variansi yang dilakukan dengan terlebih dahulu menghitung nilai rerata P.

$$\begin{aligned}\mu &= \frac{(0 + 0 + 0,17 + 0,17 + \dots + 0,33)}{1444} \\ &= \frac{1,35}{1444} \\ &= 0,000935\end{aligned}$$

$$\begin{aligned}\text{Variance} &= ((0 - 0,000935)^2 * 0) + ((30 - 0,000935)^2 * 0,17) + \dots + ((32 - 0,000935)^2 * 0,17) \\ &= 0 + 152,9905 + \dots + 174,0698 \\ &= 2399,721\end{aligned}$$

$$\begin{aligned}\text{Entropy} &= -((0,17) * \log(0,17)) + \dots + (-(0,33) * \log(0,33)) \\ &= 0 + 0,1308 + \dots + 0,1589 \\ &= 0,8129\end{aligned}$$

d) Sudut 135°

$$\begin{aligned}\text{Homogeneity} &= 0^2 + 0^2 + 0,25^2 + 0,25^2 + \dots + 0,25^2 \\ &= 0^2 + 0^2 + 0,00289 + 0,00289 + \dots + 0,1089 \\ &= 2\end{aligned}$$

$$\begin{aligned}\text{Contrast} &= ((0 - 0)^2 * 0) + ((0 - 0)^2 * 0) + ((30 - 33)^2 * 0,25) + ((35 - 38)^2 * 0,25) + \dots + ((34 - 37)^2 * 0,25) \\ &= 0 + 0 + 2,25 + 2,25 + \dots + 2,25 \\ &= 15,75\end{aligned}$$

Perhitungan correlation dilakukan dengan terlebih dahulu menghitung rerata nilai P<sub>x</sub> dan P<sub>y</sub>, serta standar deviasi nilai P<sub>x</sub> dan P<sub>y</sub>.

$$\begin{aligned}\mu_x &= (0 * 0) + (0 * 0) + (35 * 0,25) + (38 * 0,25) + \dots + (34 * 0,25) \\ &= (0) + (0) + (8,75) + (9,5) + \dots + (8,5) \\ &= 34,25\end{aligned}$$

$$\begin{aligned}\mu_y &= (0 * 0) + (0 * 0) + (33 * 0,17) + (38 * 0,17) + \dots + (32 * 0,33) \\ &= (0) + (0) + (8,25) + (9,5) + \dots + (8) \\ &= 35\end{aligned}$$

$$\begin{aligned}\sigma_x &= (0 * \sqrt{((0 - 34,25)^2)}) + (0 * \sqrt{((0 - 34,25)^2)}) + (0,25 * \sqrt{((35 - 34,25)^2)}) + (0,25 * \sqrt{((38 - 34,25)^2)}) + \dots + (0,25 * \sqrt{((34 - 34,25)^2)}) \\ &= (0) + (0) + (0,1875) + (0,9375) + \dots + (0,0625) \\ &= 2,25\end{aligned}$$

$$\begin{aligned}\sigma_y &= (0 * \sqrt{((0 - 35)^2)}) + (0 * \sqrt{((0 - 35)^2)}) + (0,25 * \sqrt{((38 - 35)^2)}) + (0,25 * \sqrt{((33 - 35)^2)}) + \dots + (0,25 * \sqrt{((32 - 35)^2)}) \\ &= (0) + (0) + (0,75) + (0,5) + \dots + (0,75) \\ &= 2,5\end{aligned}$$



$$\begin{aligned}
 \text{Correlation} &= \left( \frac{(((0 - 0) * 0) - (34,25 - 35))}{2,25 * 2,5} \right) \\
 &+ \left( \frac{(((30 - 33) * 0,25) - (34,25 - 35))}{2,25 * 2,5} \right) + \dots \\
 &+ \left( \frac{(((35 - 38) * 0,25) - (34,25 - 35))}{2,25 * 2,5} \right) \\
 &= (0) + (0) + \dots + (0,267) \\
 &= 0,667
 \end{aligned}$$

Selanjutnya, menghitung nilai variansi yang dilakukan dengan terlebih dahulu menghitung nilai rerata P.

$$\begin{aligned}
 \mu &= \frac{(0 + 0 + 0,25 + 0,25 + \dots + 0,25)}{1444} \\
 &= \frac{1,35}{1444} \\
 &= 0,00139
 \end{aligned}$$

$$\begin{aligned}
 \text{Variance} &= ((0 - 0,00139)^2 * 0) + ((30 - 0,00139)^2 * 0,25) + \dots + ((35 - 0,00139)^2 * 0,25) \\
 &= 0 + 224,9792 + \dots + 306,2257 \\
 &= 2408,057
 \end{aligned}$$

$$\begin{aligned}
 \text{Entropy} &= -((0,25) * \log(0,25)) + \dots + (-((0,25) * \log(0,25))) \\
 &= 0 + 0,1505 + \dots + 0,1505 \\
 &= 1,204
 \end{aligned}$$

#### 4.2.2 Ekstraksi Warna (*Color Channel*)

Proses ekstraksi warna dengan metode *color channel* ini menggunakan 3 fitur yaitu *mean*, *skewness*, dan *standard deviation*. Perhitungan dilakukan dengan menggunakan nilai piksel channel warna RGB sehingga untuk satu citra latih akan didapatkan 9 fitur. Gambar 4.6 berikut menunjukkan nilai piksel RGB citra sampel hasil *pre-processing* berukuran 3x3 yang akan digunakan dalam proses ekstraksi nilai fitur warna pada citra latih.

Red				Green				Blue			
x,y	0	1	2	x,y	0	1	2	x,y	0	1	2
0	76	72	76	0	17	11	17	0	21	18	23
1	69	75	71	1	19	22	16	1	22	27	22
2	67	68	66	2	21	23	17	2	24	26	21

Gambar 4.20 Nilai piksel RGB citra sampel tersegmentasi berukuran 3x3

Untuk mendapatkan ketiga fitur tersebut, terdapat tiga persamaan yang akan digunakan. Perhitungan berikut dapat membantu memberikan gambaran terkait manualisasi ekstraksi warna ini.

a) *Channel Red*

$$\text{mean} = \frac{76 + 72 + 76 + 69 + \dots + 66}{9}$$

$$= \frac{640}{9}$$

$$= 71,1$$

Selanjutnya, adalah menghitung nilai fitur standar deviasi dengan terlebih dahulu menghitung nilai variansi.

$$\begin{aligned} \text{variance} &= ((76 - 71,1)^2) + ((72 - 71,1)^2) + \dots + ((66 - 71,1)^2) \\ &= 24,01 + 0,81 + \dots + 26,01 \\ &= 94,88 \end{aligned}$$

$$\begin{aligned} \text{stddev} &= \sqrt{94,88} \\ &= 9,74 \end{aligned}$$

$$\begin{aligned} \text{skewness} &= \frac{((76 - 71,1)^3) + ((72 - 71,1)^3) + \dots + ((66 - 71,1)^3)}{9,74} \\ &= 12,079 + 0,0749 + \dots + (-13,6192) \\ &= 19,23737 \end{aligned}$$

b) Channel Green

$$\begin{aligned} \text{mean} &= \frac{17 + 11 + 17 + 19 + \dots + 17}{9} \\ &= \frac{163}{9} \\ &= 18,1 \end{aligned}$$

Selanjutnya, adalah menghitung nilai fitur standar deviasi dengan terlebih dahulu menghitung nilai variansi.

$$\begin{aligned} \text{variance} &= ((17 - 18,1)^2) + ((11 - 18,1)^2) + \dots + ((17 - 18,1)^2) \\ &= 1,21 + 50,41 + \dots + 1,21 \\ &= 105,68 \end{aligned}$$

$$\begin{aligned} \text{stddev} &= \sqrt{105,68} \\ &= 10,28 \end{aligned}$$

$$\begin{aligned} \text{skewness} &= \frac{((17 - 18,1)^3) + ((11 - 18,1)^3) + \dots + ((17 - 18,1)^3)}{10,28} \\ &= (-0,1295) + (-34,8162) + \dots + (-0,12947) \\ &= -16,3179 \end{aligned}$$

c) Channel Blue

$$\begin{aligned} \text{mean} &= \frac{21 + 18 + 23 + 22 + \dots + 21}{9} \\ &= \frac{204}{9} \\ &= 22,667 \end{aligned}$$

Selanjutnya, adalah menghitung nilai fitur standar deviasi dengan terlebih dahulu menghitung nilai variansi.

$$\begin{aligned} \text{variance} &= ((21 - 22,67)^2) + ((18 - 22,67)^2) + \dots + ((21 - 22,67)^2) \\ &= 2,779 + 21,7809 + \dots + 2,779 \\ &= 57,221 \end{aligned}$$

$$\begin{aligned} \text{stddev} &= \sqrt{57,221} \\ &= 7,5645 \end{aligned}$$



$$\begin{aligned}
 skewness &= \frac{((21 - 22,67)^3) + ((18 - 22,67)^3) + \dots + ((21 - 22,67)^3)}{10,28} \\
 &= (-0,6123) + (-13,438) + \dots + (-0,6124) \\
 &= 1,8383
 \end{aligned}$$

### 4.2.3 Klasifikasi K-NN

Proses klasifikasi dilakukan apabila nilai ekstraksi fitur tekstur dan warna telah berhasil didapat. Berdasarkan nilai-nilai fitur warna dan tekstur tersebut citra yang ingin diidentifikasi jenisnya dapat diketahui. K-NN akan langsung mengkategorikan citra kueri yang sudah dikelompokkan sebelumnya sehingga jika nilai fitur tekstur dan warna pada citra kueri cocok maka, citra kueri tersebut berhasil diidentifikasi atau berhasil dikenali jenisnya. Berikut langkah-langkah melakukan klasifikasi menggunakan metode K-NN.

1. Menentukan klasifikasi data berdasarkan nilai fitur tekstur dan warna. Gambar berikut menunjukkan data pengelompokan fitur beserta nilai fitur tekstur dan warnanya untuk empat jenis citra makanan. Pada manualisasi ini, jenis citra makanan yang digunakan yaitu dua buah donat dan 2 dua buah apel. Empat citra makanan ini disesuaikan dengan jumlah  $k$  yang digunakan, sehingga pada akhir perhitungan dapat dengan mudah diidentifikasi citra jenis makanan yang memiliki kesamaan paling banyak dengan citra makanan pada nilai ekstraksi fitur tesktur dan warna pada citra kueri. Tabel 4.1 menunjukkan data nilai ekstraksi fitur tekstur dan warna pada dua jenis citra makanan.

**Tabel 4.1 Tabel Ekstraksi Fitur Tesktur dan Warna Citra Makanan**

Ekstraksi Fitur	Donat 1	Donat 2	Apel 1	Apel 2	Citra Kueri
Homogeneity (0°)	5,2	5,3	5,4	5,5	5,6
Contrast (0°)	7,1	7,2	7,3	7,4	7,5
Correlation (0°)	1,2	1,3	1,4	1,5	1,6
Variance (0°)	3,4	3,5	3,6	3,7	3,8
Entrophy (0°)	5,7	5,8	5,9	5,10	5,11
Homogeneity (45°)	8,8	8,9	8,10	8,11	8,12
Contrast (45°)	1,2	1,3	1,4	1,5	1,6
Correlation (45°)	8,9	8,10	8,11	8,12	8,13
Variance (45°)	10,9	10,10	10,11	10,12	10,13
Entrophy (45°)	1,6	1,7	1,8	1,9	1,10
Homogeneity (90°)	0,03	0,04	0,05	0,06	0,07
Contrast (90°)	2,7	2,8	2,9	2,10	2,11
Correlation (90°)	8,0	8,1	8,2	8,3	8,4
Variance (90°)	9,2	9,3	9,4	9,5	9,6
Entrophy (90°)	0,01	0,02	0,03	0,04	0,05
Homogeneity (135°)	9,3	9,4	9,5	9,6	9,7
Contrast (135°)	3,3	3,4	3,5	3,6	3,7
Correlation (135°)	7,6	7,7	7,8	7,9	7,10
Variance (135°)	5,2	5,3	5,4	5,5	5,6

Entropy (135°)	8,1	8,2	8,3	8,4	8,5
Mean (Red)	7,0	7,1	7,2	7,3	7,4
Standar Deviation (Red)	4,0	4,1	4,2	4,3	4,4
Skewness (Red)	5,5	5,6	5,7	5,8	5,9
Mean (Green)	7,9	7,10	7,11	7,12	7,13
Standar Deviation (Green)	9,0	9,1	9,2	9,3	9,4
Skewness (Green)	10,2	10,3	10,4	10,5	10,6
Mean (Blue)	8,3	8,4	8,5	8,6	8,7
Standar Deviation (Blue)	5,2	5,3	5,4	5,5	5,6
Skewness (Blue)	0,09	0,10	0,11	0,12	0,13

- Menentukan parameter  $k$  (jumlah tetangga paling dekat) misalnya  $k=3$ .
- Menghitung kuadrat jarak euclid (*query distance*) pada masing-masing objek terhadap data sampel yang diberikan. Jika diketahui suatu citra kueri yang memiliki informasi sebagai berikut:

$$\begin{aligned} \text{Donat 1} &= ((5.2 - 5.6)^2) + ((7.1 - 7.5)^2) + \dots + ((0.09 - 0.13)^2) \\ &= 6.8 \end{aligned}$$

$$\begin{aligned} \text{Donat 2} &= ((5.3 - 5.6)^2) + ((7.2 - 7.5)^2) + \dots + ((0.10 - 0.13)^2) \\ &= 4.31 \end{aligned}$$

$$\begin{aligned} \text{Apel 1} &= ((5.4 - 5.6)^2) + ((7.3 - 7.5)^2) + \dots + ((0.11 - 0.13)^2) \\ &= 2.95 \end{aligned}$$

$$\begin{aligned} \text{Apel 2} &= ((5.5 - 5.6)^2) + ((7.4 - 7.5)^2) + \dots + ((0.12 - 0.13)^2) \\ &= 1.46 \end{aligned}$$

- Setelah menghitung jarak Euclidean antara citra kueri dengan citra pada basis data, tahap selanjutnya adalah mengelompokkan citra ke dalam kelompok yang mempunyai jarak euclid terkecil. Tabel 4.2 menunjukkan perbandingan jarak antar kategori kelas. Kemudian, mengurutkannya dari jarak terkecil. Penetapan nilai  $k=3$  menyebabkan citra Donat 1 tidak termasuk ke dalam kategorisasi K-NN sehingga dianggap tidak termasuk K-NN.

**Tabel 4.2 Tabel Perbandingan Jarak Antar Kategori Kelas**

No	Jenis Makanan	Square Distance to Query Distance	Jarak Terkecil	Tetangga Terdekat (Y/N)
1	Donat 1	6,80	4	N
2	Donat 2	4,31	3	Y
3	Apel 1	2,95	2	Y
4	Apel 2	1,46	1	Y

- Berdasarkan tabel pada Gambar dapat di analisis bahwa terdapat 3 citra yang termasuk ke tetangga terdekat K-NN dan 1 yang tidak termasuk ( $3 > 1$ ), sehingga dapat diketahui bahwa citra makanan tersebut diidentifikasi memiliki kesamaan dengan citra kueri, sehingga perlu dilakukan perhitungan frekuensi jenis citra makanan untuk mendapatkan satu jenis citra yang memiliki kesamaan terhadap citra kueri. Tabel 4.3 berikut menunjukkan ketiga citra jenis makanan yang mendekati citra kueri.

**Tabel 4.3 Tabel Citra Makanan dengan Jarak Terdekat dengan Citra Kueri**

No	Jenis Makanan	Square Distance to Query Distance
1	Donat 2	4,31
2	Apel 1	2,95
3	Apel 2	1,46

Berdasarkan Tabel 4.3 tersebut terdapat 2 citra apel dan 1 citra donat. Hal ini dapat disimpulkan bahwa citra kueri adalah citra apel karena citra apel memiliki frekuensi yang lebih besar ( $2 > 1$ ) dibandingkan dengan citra donat.

### 4.3 Perancangan Skenario Pengujian

Pada penelitian ini pengujian dilakukan berdasarkan algoritme yang digunakan untuk menghasilkan *output* berupa kemiripan citra pada data uji dan data latih. Terdapat 4 pengujian yang dilakukan yaitu pengujian nilai  $k$ , pengujian dengan menggunakan fitur warna, pengujian dengan menggunakan fitur tekstur, dan pengujian menggunakan fitur warna dan tekstur. Berikut penjabaran lengkap skenario pengujian yang dilakukan pada penelitian ini:

1. Pengujian nilai  $k$

Pengujian nilai  $k$  dilakukan dengan menguji keakurasian tetangga terdekat pada data uji terhadap data latih. Nilai  $k$  yang digunakan yaitu 3, 5, 7, 9, 11, 13, dan 15. Tujuan dari pengujian ini adalah untuk mendapatkan kombinasi nilai  $k$  dengan akurasi terbaik sesuai dengan jumlah kelas pada data uji. Data uji yang digunakan pada pengujian yaitu dataset dengan jumlah data yang sama pada setiap kelas (seimbang) dan jumlah data yang tidak sama jumlahnya pada setiap kelas (tidak seimbang). Rancangan Pengujian ini ditunjukkan pada Tabel 4.4.

**Tabel 4.4 Pengujian nilai  $k$  pada data seimbang dan data tidak seimbang**

Nilai $k$	Data Seimbang	Data Tidak Seimbang
3		
5		
7		
9		
11		
13		
15		

2. Pengujian ekstraksi fitur

Pengujian ini dilakukan dengan menguji akurasi klasifikasi jenis makanan dengan hanya menggunakan salah satu jenis ekstraksi fitur yaitu fitur warna atau fitur tekstur, serta menggunakan kedua jenis ekstraksi fitur. Dataset yang digunakan sama dengan dataset yang digunakan pada pengujian sebelumnya yaitu data uji seimbang dan data uji tidak seimbang. Dengan demikian, pengujian ini akan mendapatkan hasil berupa kombinasi akurasi



ekstraksi fitur terhadap dua jenis data uji tersebut. Tabel 4.23 berikut menunjukkan rancangan pengujian menggunakan fitur warna.

**Tabel 4.5 Pengujian ekstraksi fitur**

Ekstraksi Fitur	Jumlah Sub Fitur	Akurasi	
		Data seimbang	Data tidak seimbang
Warna	9		
Tekstur	20		
Kombinasi	29		

3. Pengujian perhitungan jarak tetangga terdekat

Pengujian ini dilakukan dengan cara membandingkan dua metode perhitungan jarak tetangga terdekat yaitu *Euclidean distance* dan *manhattan distance*. Hasil perbandingan penggunaan metode tersebut digunakan untuk mengetahui pengaruh penggunaan metode perhitungan jarak tetangga terdekat dengan hasil akurasi klasifikasi jenis makanan. Selain itu, pengujian ini juga menggunakan nilai  $k$  sebagai parameter untuk mengetahui hubungan antara penerapan 2 metode perhitungan jarak tetangga terdekat dengan akurasi hasil kalsifikasi berdasarkan nilai  $k$  yang sudah ditentukan. Tabel 4.25 dan Tabel 4.26 menunjukkan rancangan pengujian ini.

**Tabel 4.6 Pengujian perhitungan jarak terdekat**

Metode jarak terdekat	Akurasi					
	Data seimbang			Data tidak seimbang		
	Warna	Tekstur	Kombinasi	Warna	Tekstur	Kombinasi
Euclidean Distance						
Manhattan Distance						



## BAB 5 IMPLEMENTASI

Pada Bab 5 ini akan dibahas mengenai penerapan sistem berdasarkan perancangan sistem yang sudah dibuat sebelumnya pada Bab Analisis dan Perancangan. Bab Implementasi sendiri terdiri atas beberapa sub bab yaitu lingkungan implementasi, batasan implementasi, dan implementasi aplikasi. Berikut akan dibahas satu persatu terkait sub bab- sub bab tersebut di atas.

### 5.1 Lingkungan Implementasi

#### 5.1.1 Lingkungan *Hardware*

Pengimplementasian sistem klasifikasi citra jenis makanan ini menggunakan beberapa spesifikasi perangkat keras yang dijelaskan sebagai berikut:

1. *Processor*: Intel Core i5-7200U, up to 3.16GHz
2. *Memory*: 4 GB
3. *VGA*: NVIDIA GEFORCE 930MX
4. *Harddisk* 1 TB
5. Monitor 14 inch
6. Keyboard
7. Mouse

#### 5.1.2 Lingkungan *Software*

Pengimplementasian sistem klasifikasi citra jenis makanan ini menggunakan beberapa spesifikasi perangkat lunak yang dijelaskan sebagai berikut:

1. *Operating System* Microsoft Windows 10 Pro
2. Microsoft Office Word 2010 sebagai *software* yang membantu dalam pembuatan laporan dan dokumentasi penelitian.
3. Microsoft Office Excel 2010 sebagai *software* yang membantu dalam pembuatan manualisasi sistem, penyimpanan nilai hasil ekstraksi fitur, dan pengumpulan data hasil pengujian
4. Microsoft Office Power Point 2010 sebagai *software* yang membantu dalam pembuatan *slide* presentasi penelitian.
5. Pycharm versi 2018.2.4 (menggunakan numpy, pandas, cv2, dan os) sebagai *Integrated Development Environment* (IDE) yang membantu dalam pengimplemetasi program pada sistem yang sudah di rancang.

### 5.2 Batasan Implementasi

Batasan masalah pada pengimplementasian klasifikasi citra jenis makanan ini dijabarkan sebagai berikut:

1. Data yang digunakan dalam penelitian ini terdiri atas beberapa jenis data yang terbagi atas 2 kategori yaitu data latih dan data uji. Pada proses *training*

total data yang digunakan sebanyak 616 data. Data-data tersebut terdiri atas 150 data latih seimbang dan 466 data latih tidak seimbang. Sedangkan pada proses *testing* total data yang digunakan sebanyak 114 data. Data-data tersebut terdiri atas 45 data latih seimbang dan 69 data latih tidak seimbang. Total citra yang digunakan dalam penelitian ini adalah sebanyak 730 citra.

2. Metode *pre-processing* yang digunakan untuk dapat mensegmentasikan citra sehingga didapatkan fokus citra adalah *Gaussian Blur*, Transformasi RGB ke HSV, *Grayscale*, *Transformasi Biner*, dan *Opening*.
3. Ekstraksi fitur yang digunakan dalam penelitian ini adalah *color channel* sebagai metode untuk ekstraksi warna dan GLCM sebagai metode untuk ekstraksi tekstur. Pada *color channel* fitur yang digunakan adalah sebanyak 3 fitur sedangkan GLCM menggunakan 5 fitur.
4. Ukuran citra yang digunakan yaitu 3096 x 4128 piksel yang pada proses segmentasi dilakukan *resizing* sehingga ukuran citra dataset menjadi 200 x 200 piksel. Citra tersebut kemudian digunakan pada proses ekstraksi fitur *color channel* dan GLCM.
5. Klasifikasi citra pada penelitian ini menggunakan K-NN (*K-Nearest Neighbor*) serta menggunakan *Euclidean distance* sebagai metode perhitungan tetangga terdekat.

## 5.3 Implementasi Aplikasi

### 5.3.1 Pre-Processing

*Pre-Processing* merupakan tahapan pertama yang harus dilakukan untuk dapat mengklasifikasikan citra yang bertujuan agar proses pengambilan nilai ekstraksi fitur menjadi lebih mudah. Citra hasil segmentasi ini nantinya akan digunakan sebagai citra sumber untuk mendapat nilai ekstraksi fitur warna dan tekstur. Implementasi kode program untuk *pre-processing* ini adalah dengan memanggil fungsi `preprocessing()` seperti pada *Sourcecode* 5.1 berikut.

```

1 def preprocessing(self, image, path, filenames):
2     img1 = cv2.resize(image, (200,200))
3     img2 = cv2.resize(image, (200,200))
4     gaussianBlur = cv2.GaussianBlur(img2, (5, 5), 30)
5     image = cv2.cvtColor(gaussianBlur, cv2.COLOR_BGR2HSV)
6     ret, mask = cv2.threshold(image, 50, 255,
7 cv2.THRESH_BINARY)
8     img2gray = cv2.cvtColor(mask, cv2.COLOR_BGR2GRAY)
9     ret, mask1 = cv2.threshold(img2gray, 110, 255,
10 cv2.THRESH_BINARY)
11     kernel = np.ones((3,3), np.uint8)
12     opening = cv2.morphologyEx(mask1, cv2.MORPH_OPEN, kernel,
13 iterations =3)
14     img1_bg = cv2.bitwise_and(img1, img1, mask = opening)
15     cv2.imwrite(os.path.join(path, filenames), img1_bg)

```

#### Kode Program 5.1 Implementasi fungsi `preprocessing()`

Kode program di atas dapat dijabarkan sebagai berikut:

1. Baris 1 merupakan pendeklarasian fungsi preprocessing() dengan paramater self, image, path, dan filenames sebagai inputan fungsi.
2. Baris 2-3 merupakan proses *resizing image* menjadi citra berukuran 200 x 200 piksel.
3. Baris 4 merupakan rumus yang dipanggil dari *library* cv2 yang mengubah citra asli menjadi blur menggunakan fungsi .GaussianBlur sehingga menghasilkan matriks 5x5 sebanyak 30 iterasi.
4. Baris 5 mengubah citra yang sudah di blur tadi menjadi HSV dari RGB.
5. Baris 6-7 mengubah warna citra HSV menjadi citra biner dengan menetapkan nilai 60 sebagai ambang batas. Jika nilai piksel lebih dari nilai ambang batas maka nilai piksel berubah menjadi 1 dan sebaliknya.
6. Baris 8 mengubah citra HSV menjadi citra grayscale yang dipanggil dari *library* cv2 menggunakan fungsi .cvtColor.
7. Baris 9-10 melakukan pembagian area dimana area piksel yang melebihi nilai ambang batas yaitu 110, maka nilai piksel berubah menjadi 1 dan sebaliknya.
8. Baris 11 menentukan kernel matriks yaitu 3x3
9. Baris 12-13 menghapus noise menggunakan metode opening yang dipanggil dari *library* cv2 yang mengubah menggunakan fungsi .morphologyEx.
10. Baris 14 mengubah citra menjadi citra bitwise yang dipanggil dari *library* cv2 yang mengubah menggunakan fungsi .bitwise.
11. Baris 15 menyimpan citra hasil segmentasi ke folder yang sudah ditentukan.

### 5.3.2 Color Channel

#### 5.3.2.1 Implementasi Ekstraksi Fitur Mean

```

1 def mean(image):
2     mean_b,mean_g,mean_r = image[:, :, 0],
3     image[:, :, 1], image[:, :, 2]
4     b1, g1, r1=mean_b.ravel(), mean_g.ravel(), mean_r.ravel()
5     rerata_b = sum(b1) / len(b1)
6     rerata_g = sum(g1) / len(g1)
7     rerata_r = sum(r1) / len(r1)
8     return rerata_b, rerata_g, rerata_r

```

**Kode Program 5.2 Implementasi fungsi mean()**

Kode program di atas dapat dijabarkan sebagai berikut:

1. Baris 1 merupakan pendeklarasian fungsi mean() dengan paramater image sebagai inputan fungsi.
2. Baris 2-3 merupakan deklarasi array nilai *Red*, *Green*, dan *Blue* yang disimpan dalam variabel `mean_b`, `mean_g`, dan `mean_r`.
3. Baris 4 merupakan deklarasi pengumpulan nilai-nilai *Red*, *Green*, dan *Blue* yang masing-masing disimpan dalam variabel `b1`, `g1`, dan `r1`.
4. Baris 5 merupakan perhitungan rata-rata nilai *Blue* dalam citra yang disimpan dalam variabel `rerata_b`.
5. Baris 6 merupakan perhitungan rata-rata nilai *Green* dalam citra yang disimpan dalam variabel `rerata_g`.

6. Baris 7 merupakan perhitungan rata-rata nilai *Red* dalam citra yang disimpan dalam variabel `rerata_r`.
7. Baris 8 merupakan pengembalian nilai variabel `rerata_b`, `rerata_g`, dan `rerata_r`.

### 5.3.2.2 Implementasi Ekstraksi Fitur *Skewness*

```

1 def skewness(image):
2     skew_b, skew_g, skew_r = image[:, :, 0], image[:, :, 1],
3     image[:, :, 2]
4     b2, g2, r2 = skew_b.ravel(), skew_g.ravel(),
5     skew_r.ravel()
6     mean_b, mean_g, mean_r =
7     mean(image)[0], mean(image)[1], mean(image)[2]
8     b2 = (b2 - mean_b)**3
9     g2 = (g2 - mean_g)**3
10    r2 = (r2 - mean_r)**3
11    b = sum(b2) / len(b2)
12    g = sum(g2) / len(g2)
13    r = sum(r2) / len(r2)
14    skew_b = np.cbrt(b)
15    skew_g = np.cbrt(g)
16    skew_r = np.cbrt(r)
17    return skew_b, skew_g, skew_r

```

#### Kode Program 5.3 Implementasi fungsi `skewness()`

Kode program di atas dapat dijabarkan sebagai berikut:

1. Baris 1 merupakan pendeklarasian fungsi `skewness()` dengan paramater `image` sebagai inputan fungsi.
2. Baris 2-3 merupakan deklarasi array nilai *Red*, *Green*, dan *Blue* yang disimpan dalam variabel `skew_b`, `skew_g`, dan `skew_r`.
3. Baris 4-5 merupakan deklarasi pengumpulan nilai-nilai *Red*, *Green*, dan *Blue* yang masing-masing disimpan dalam variabel `b2`, `g2`, dan `r2`.
4. Baris 6-7 merupakan deklarasi indeks array nilai-nilai fitur terhadap warna *Red*, *Green*, dan *Blue* yang disimpan dalam variabel `mean_b`, `mean_g`, dan `mean_r`.
5. Baris 8 merupakan perhitungan selisih rata-rata nilai *Blue* dalam citra dengan rata-rata jumlah nilai piksel *Blue* kemudian di pangkatkan 3 yang disimpan dalam variabel `b2`.
6. Baris 9 merupakan perhitungan selisih rata-rata nilai *Green* dalam citra dengan rata-rata jumlah nilai piksel *Green* kemudian di pangkatkan 3 yang disimpan dalam variabel `g2`.
7. Baris 10 merupakan perhitungan selisih rata-rata nilai *Red* dalam citra dengan rata-rata jumlah nilai piksel *Red* kemudian di pangkatkan 3 yang disimpan dalam variabel `r2`.
8. Baris 11 menghitung jumlah piksel pada variabel `b2` dibagi dengan panjang piksel `b2` yang hasilnya disimpan dalam variabel `b`.
9. Baris 12 menghitung jumlah piksel pada variabel `g2` dibagi dengan panjang piksel `g2` yang hasilnya disimpan dalam variabel `g`.

10. Baris 13 menghitung jumlah piksel pada variabel `r2` dibagi dengan panjang piksel `r2` yang hasilnya disimpan dalam variabel `r`.
11. Baris 14 menghitung akar pangkat tiga dari nilai *skewness* pada piksel *Blue* yang hasilnya disimpan pada variabel `skew_b`.
12. Baris 15 menghitung akar pangkat tiga dari nilai *skewness* pada piksel *Green* yang hasilnya disimpan pada variabel `skew_g`.
13. Baris 16 menghitung akar pangkat tiga dari nilai *skewness* pada piksel *Red* yang hasilnya disimpan pada variabel `skew_r`.
14. Baris 17 merupakan pengembalian nilai variabel `skew_b`, `skew_g`, dan `skew_r`.

### 5.3.2.3 Implementasi Ekstraksi Fitur Standar Deviasi

```

1 def stdev (image):
2     std_b, std_g, std_r = image[:, :, 0], image[:, :, 1],
3     image[:, :, 2]
4     b3, g3, r3 = std_b.ravel(), std_g.ravel(), std_r.ravel()
5     mean_b, mean_g, mean_r = mean(image)[0], mean(image)[1],
6     mean(image)[2]
7     b3 = (b3 - mean_b) ** 2
8     g3 = (g3 - mean_g) ** 2
9     r3 = (r3 - mean_r) ** 2
10    b = sum(b3) / len(b3)
11    g = sum(g3) / len(g3)
12    r = sum(r3) / len(r3)
13    std_b = np.cbrt(b)
14    std_g = np.cbrt(g)
15    std_r = np.cbrt(r)
16    return std_b, std_g, std_r

```

**Kode Program 5.4 Implementasi fungsi stdev()**

Kode program di atas dapat dijabarkan sebagai berikut:

1. Baris 1 merupakan pendeklarasian fungsi `stdev()` dengan paramater `image` sebagai inputan fungsi.
2. Baris 2-3 merupakan deklarasi array nilai *Red*, *Green*, dan *Blue* yang disimpan dalam variabel `std_b`, `std_g`, dan `std_r`.
3. Baris 4 merupakan deklarasi pengumpulan nilai-nilai *Red*, *Green*, dan *Blue* yang masing-masing disimpan dalam variabel `b3`, `g3`, dan `r3`.
4. Baris 5-6 merupakan deklarasi indeks array nilai-nilai fitur terhadap warna *Red*, *Green*, dan *Blue* yang disimpan dalam variabel `mean_b`, `mean_g`, dan `mean_r`.
5. Baris 7 merupakan perhitungan selisih rata-rata nilai *Blue* dalam citra dengan rata-rata jumlah nilai piksel *Blue* kemudian di pangkatkan 2 yang disimpan dalam variabel `b3`.
6. Baris 8 merupakan perhitungan selisih rata-rata nilai *Green* dalam citra dengan rata-rata jumlah nilai piksel *Green* kemudian di pangkatkan 2 yang disimpan dalam variabel `g3`.
7. Baris 9 merupakan perhitungan selisih rata-rata nilai *Red* dalam citra dengan rata-rata jumlah nilai piksel *Red* kemudian di pangkatkan 2 yang disimpan dalam variabel `r3`.

8. Baris 10 menghitung jumlah piksel pada variabel `b3` dibagi dengan panjang piksel `b3` yang hasilnya disimpan dalam variabel `b`.
9. Baris 11 menghitung jumlah piksel pada variabel `g3` dibagi dengan panjang piksel `g3` yang hasilnya disimpan dalam variabel `g`.
10. Baris 12 menghitung jumlah piksel pada variabel `r3` dibagi dengan panjang piksel `r3` yang hasilnya disimpan dalam variabel `r`.
11. Baris 13 menghitung akar pangkat tiga dari nilai standar deviasi pada piksel *Blue* yang hasilnya disimpan pada variabel `std_b`.
12. Baris 14 menghitung akar pangkat tiga dari nilai standar deviasi pada piksel *Green* yang hasilnya disimpan pada variabel `std_g`.
13. Baris 15 menghitung akar pangkat tiga dari nilai standar deviasi pada piksel *Red* yang hasilnya disimpan pada variabel `std_r`.
14. Baris 16 merupakan pengembalian nilai variabel `std_b`, `std_g`, dan `std_r`.

### 5.3.3 Gray Level CO-OCCURENCE MATRIX (GLCM)

```

1 def matriksglcm(img, sudut, d):
2     image2 = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
3     glcmimg = np.array(image2).astype(np.uint8)
4     t, l = glcmimg.shape
5     G = 256
6     GLCM = np.zeros((G,G))
7     if sudut == 0:
8         for b in range(t-d):
9             for k in range(l-d):
10                if (glcmimg[b,k] != 0) and (glcmimg[b,k+d]
11                != 0):
12                    t1 = glcmimg[b,k]
13                    t2 = glcmimg[b, k+d]
14                    GLCM[t1,t2] += 1
15            elif sudut == 45:
16                for b in range(t-d):
17                    for k in range(l-d):
18                        if (glcmimg[b,k] != 0) and (glcmimg[b-1,k+d]
19                        != 0):
20                            t1 = glcmimg[b,k]
21                            t2 = glcmimg[b-d, k+d]
22                            GLCM[t1,t2] += 1
23            elif sudut == 90:
24                for b in range(t - d):
25                    for k in range(l - d):
26                        if (glcmimg[b, k] != 0) and (glcmimg[b - d,k
27                        + d] != 0):
28                            t1 = glcmimg[b, k]
29                            t2 = glcmimg[b - d, k]
30                            GLCM[t1, t2] += 1
31            elif sudut == 135:
32                for b in range(t - d):
33                    for k in range(l - d):
34                        if (glcmimg[b, k] != 0) and (glcmimg[b - d,k
35                        - d] != 0):
36                            t1 = glcmimg[b, k]

```

```

34         t2 = glcmimg[b - d, k-d]
35         GLCM[t1, t2] += 1
36     Transpose = np.zeros((G,G))
37     totalPiksel = 0
38     for baris in range (G-1):
39         for kolom in range(G-1):
40             Transpose[baris,kolom] = GLCM[baris,kolom]
41             +GLCM[kolom,baris]
42             totalPiksel+= Transpose[baris,kolom]
43     normalisasi = np.zeros((G,G))
44     normalisasi = Transpose/totalPiksel
45     return normalisasi

```

### Kode Program 5.5 Implementasi matriksglcm()

Kode program di atas dapat dijabarkan sebagai berikut:

1. Baris 1 merupakan pendeklarasian fungsi matriksglcm() dengan paramater img, sudut, dan  $d$  sebagai inputan fungsi.
2. Baris 2 mengubah citra tersegmentasi (img) menjadi citra grayscale.
3. Baris 3 mengubah citra tersegmentasi (img) dengan tipe data Image menjadi citra array bertipe data uint8.
4. Baris 4 mendapatkan nilai panjang dan lebar dari citra inputan.
5. Baris 5 merupakan inialisasi variabel G dengan nilai 256.
6. Baris 6 inialisasi variabel GLCM yang memiliki ukuran yang sama pada citra tersegmentasi (img)
7. Baris 7-13 merupakan perhitungan matriks awal pada GLCM dengan parameter arah sama dengan 0 derajat.
8. Baris 14-20 merupakan perhitungan matriks awal pada GLCM dengan parameter arah sama dengan 45 derajat.
9. Baris 21-28 merupakan perhitungan matriks awal pada GLCM dengan parameter arah sama dengan 90 derajat.
10. Baris 29-35 merupakan perhitungan matriks awal pada GLCM dengan parameter arah sama dengan 135 derajat.
11. Baris 36 inialisasi variabel transpose yang berisi parameter ukuran citra tersegmentasi
12. Baris 37-41 merupakan looping piksel yang menjumlahkan matriks awal glcm dengan matriks transpose kemudian di simpan dalam variabel Transpose
13. Baris 42-43 matriks pada variabel Transpose tadi kemudian di lakukan proses normalisasi dengan rumus variabel Transpose dibagi dengan banyaknya keseluruhan piksel pada array Transpose.
14. Baris 44 pengembalian nilai variabel normalisasi.

#### 5.3.3.2 Implementasi Ekstraksi Fitur GLCM

```

1 def ekstraksiFitur(arr) :
2     fitur = np.zeros(5,np.double)
3     G = 256
4     #1. Contrast
5     contrast = 0
6     for i in range(G - 1):
7         for j in range(G - 1):

```

```

8         contrast += (((i-j)**2) * arr[i,j])
9     fitur[0] = contrast
10    # 2. Correlation
11    ux = 0
12    uy = 0
13    std_x = 0
14    std_y = 0
15    correlation = 0
16    for i in range(G-1):
17        for j in range(G-1):
18            ux += i*arr[i,j]
19            uy += j * arr[i, j]
20    for i in range(G - 1):
21        for j in range(G - 1):
22            std_x += arr[i,j] * ((i-ux)**2)**0.5
23            std_y += arr[i, j] * ((j - uy) ** 2) ** 0.5
24    for i in range(G - 1):
25        for j in range(G - 1):
26            correlation += ((i*j) * arr[i,j]) - (ux*uy)
27    /(std_x*std_y)
28    fitur[1] = correlation
29    # 3. Homogeneity (ASM)
30    homo = 0
31    for i in range(G - 1):
32        for j in range(G - 1):
33            homo += (arr[i,j]**2)
34    fitur[2] = homo
35    # 4. Variance
36    jumlah = 0
37    tot = 0
38    var = 0
39    for i in range(G - 1):
40        for j in range(G - 1):
41            jumlah += arr[i,j]
42            tot += 1
43    mean = jumlah/tot
44    for i in range(G - 1):
45        for j in range(G - 1):
46            var += ((i-mean)**2) * arr[i,j]
47    fitur[3]=var
48    #5. Entropy
49    entro = 0
50    for i in range(G - 1):
51        for j in range(G - 1):
52            entro += ((arr[i,j]) * np.log(arr[i,j] +
53 np.finfo(float).eps))
54    entro = entro*(-1)
55    fitur[4] = entro
56    return fitur

```

### Kode Program 5.6 Implementasi fungsi ekstraksifitur()

Kode program di atas dapat dijabarkan sebagai berikut:

1. Baris 1 merupakan pendeklarasian fungsi ekstraksifitur() dengan paramater arr sebagai inputan fungsi.
2. Baris 5-9 merupakan perhitungan ekstraksi fitur *contrast*. Hasil perhitungan kemudian disimpan ke dalam indeks array pertama.

3. Baris 11-28 merupakan perhitungan ekstraksi fitur *correlation*. Hasil perhitungan kemudian disimpan ke dalam indeks array kedua.
4. Baris 30-34 merupakan perhitungan ekstraksi fitur *homogeneity*. Hasil perhitungan kemudian disimpan ke dalam indeks array ketiga.
5. Baris 36-47 merupakan perhitungan ekstraksi fitur *variance*. Hasil perhitungan kemudian disimpan ke dalam indeks array keempat.
6. Baris 49-55 merupakan perhitungan ekstraksi fitur *entropy*. Hasil perhitungan kemudian disimpan ke dalam indeks array kelima.
7. Baris 56 mengembalikan nilai variabel fitur dengan tipe list.



## BAB 6 PENGUJIAN DAN ANALISIS

Pada Bab 6 ini akan dibahas mengenai pengujian dan analisis hasil implementasi pada program yang sudah dibuat pada Bab Implementasi. Pengujian ini dilakukan dengan menggunakan total 450 set citra yang terdiri atas 15 kelas. Pada prosesnya, pengujian dilakukan dengan mengacu pada 2 kategori data yaitu citra data seimbang dan citra data tidak seimbang masing-masing ada pada data latih dan data uji. Terdapat 3 tahap pengujian yang pada setiap pengujiannya akan disertai analisis hasil.

### 6.1 Skenario Pengujian dan Analisis

Pada Bab Analisis dan Perancangan, telah dibuat skenario dan perancangan pengujian sebagai berikut:

1. Pengujian nilai  $k$
2. Pengujian ekstraksi fitur
3. Pengujian perhitungan jarak tetangga terdekat

Skenario pengujian ini dirancang sedemikian rupa untuk mengetahui nilai akurasi terbaik sistem pengenalan citra jenis makanan terhadap 3 variabel seperti nilai  $k$ , ekstraksi fitur, serta metode perhitungan jarak tetangga terdekat. Tabel 6.1 dan Tabel 6.2 berikut menunjukkan 10 dari 15 kelas citra makanan yang berhasil diidentifikasi atau dikenali pada pengujian nilai  $k$  dengan akurasi tertinggi yaitu 90,58% untuk  $k=3$  pada kategori data seimbang.

**Tabel 6.1 Hasil identifikasi citra pada pengujian nilai  $k=3$**

Nama Citra	Citra Asli	Nama Citra	Citra Identifikasi
Donat		Donat	
Biskuat		Biskuat	
Genji Pie		Genji Pie	
Jeruk Hijau		Pisang Hijau	

**Tabel 6.2 Hasil identifikasi citra pada pengujian nilai  $k=3$  (lanjutan)**

Nama Citra	Citra Asli	Nama Citra	Citra Identifikasi
Jeruk Oranye		Jeruk Oranye	
Mie		Mie	
Milo Nugget		Biskuat	
Pisang Hijau		Pisang Hijau	
Pisang Kuning		Pisang Kuning	
Stroberi		Stroberi	

Pada skenario pengujian pertama, yaitu pengujian nilai  $k$  merupakan pengujian yang bertujuan untuk mengetahui pengaruh variasi nilai  $k$  terhadap hasil akurasi dalam pengklasifikasian citra jenis makanan pada komposisi data seimbang dan data tidak seimbang. Parameter nilai  $k$  memiliki pengaruh terhadap hasil klasifikasi jenis makanan sehingga diharapkan dengan melakukan uji kombinasi nilai  $k$  ini, akan didapatkan nilai  $k$  dengan akurasi klasifikasi terbaik berdasarkan jenis komposisi data.

Pada skenario pengujian kedua, yaitu pengujian ekstraksi fitur merupakan pengujian yang bertujuan untuk mengetahui pengaruh penggunaan fitur-fitur yaitu *color channel* dan GLCM, serta kombinasi keduanya. Parameter ekstraksi fitur ini memberikan pengaruh terhadap hasil klasifikasi karena kedua fitur tersebut menghasilkan nilai fitur yang menjadi pembeda antara jenis makanan satu dengan makanan yang lain, sehingga jika salah satu jenis ekstraksi fitur saja

atau keduanya digunakan dalam proses klasifikasi, tentu akan menghasilkan nilai akurasi yang berbeda-beda. Diharapkan dengan adanya pengujian ini akan didapatkan jenis ekstraksi fitur dengan akurasi klasifikasi terbaik untuk kedua jenis kategori data. Terakhir, pada skenario pengujian ketiga, yaitu pengujian metode perhitungan jarak tetangga terdekat. Pengujian ini dilakukan dengan membandingkan akurasi dua jenis metode perhitungan jarak tetangga terdekat yaitu *Euclidean distance* dan *manhattan distance*. Pengujian ini bertujuan untuk mengetahui metode perhitungan jarak dengan akurasi terbaik yang dapat digunakan sebagai metode perhitungan jarak tetangga terdekat pada K-NN untuk sistem klasifikasi citra jenis makanan berdasarkan jeniskategori data.

### 6.1.1 Pengujian Nilai $k$

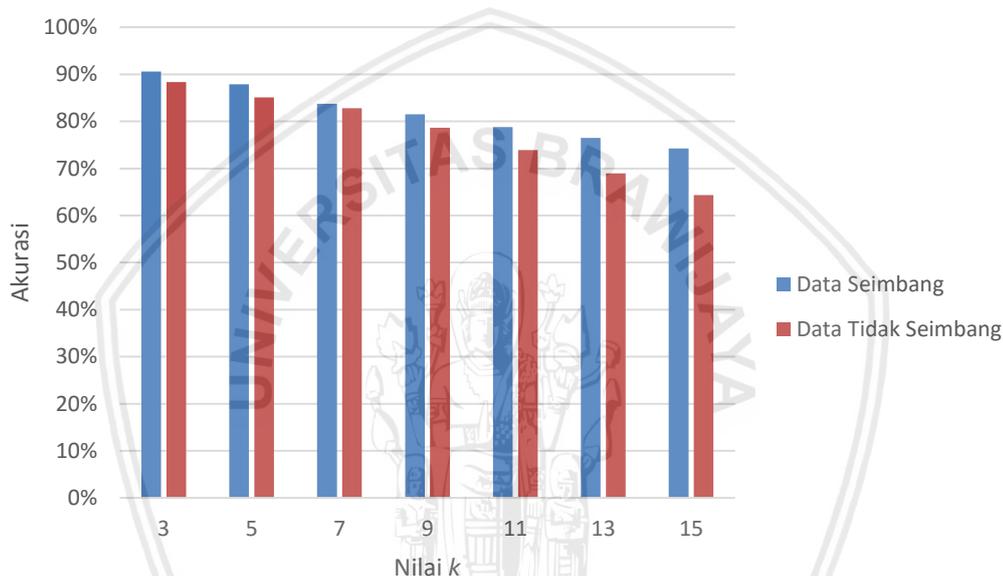
Pengujian nilai  $k$  dilakukan pada proses klasifikasi citra makanan dengan metode K-NN dimana nilai  $k$  dapat ditentukan sendiri dengan mengubah nilai variabel  $k$  pada kode program. Nilai  $k$  baru bisa ditentukan setelah mengetahui jumlah kelas datasetnya. Pada penelitian ini terdapat 15 kelas sehingga pada pengujian variasi nilai  $k$  yang digunakan adalah 3, 5, 7, 9, 11, 13, dan 15. Pengujian ini dilakukan dengan menggunakan dataset sebanyak 450 data. Data tersebut dikategorisasikan menjadi 2 jenis yaitu data seimbang dan data tidak seimbang. Data seimbang ini merupakan jenis data yang komposisi data pada setiap kelasnya memiliki jumlah yang sama, sedangkan data tidak seimbang memiliki banyak variasi jumlah data pada setiap kelasnya. Data seimbang memiliki 21 data untuk setiap kelasnya untuk data latih, sedangkan pada data uji terdapat 9 data untuk setiap kelasnya. Hasil pengujian nilai  $k$  ditunjukkan pada Tabel 6.3.

**Tabel 6.3 Pengujian nilai  $k$**

Nilai $k$	Data Seimbang	Data Tidak Seimbang
3	90,58%	88,35%
5	87,87%	85,07%
7	83,73%	82,80%
9	81,51%	78,62%
11	78,80%	73,91%
13	76,45%	68,98%
15	74,22%	64,31%

Berdasarkan hasil pengujian pada Tabel 6.3 tersebut di atas dapat diketahui bahwa nilai akurasi tertinggi ada pada nilai  $k=3$ . Pengujian nilai  $k$  ini membuktikan bahwa perubahan pada nilai  $k$  dapat memengaruhi hasil akurasi dari sistem klasifikasi citra jenis makanan ini. Grafik pada Gambar 6.1 menunjukkan bahwa semakin besar penentuan nilai  $k$ , maka hasil akurasi akan semakin menurun. Hal ini disebabkan karena beberapa faktor seperti kesesuaian data yang menyebabkan nilai validitas data latih semakin menurun jika nilai  $k$  semakin besar. Selain itu, komposisi jumlah citra setiap kelas juga memengaruhi hasil akurasi. Berdasarkan Tabel 6.3, akurasi hasil klasifikasi dengan

menggunakan data tidak seimbang lebih besar yaitu 90,58% daripada akurasi hasil klasifikasi menggunakan data seimbang dengan selisih perbandingan 2,23% pada  $k=3$ . Hal ini disebabkan karena faktor komposisi data. Dataset yang memiliki jumlah citra yang seragam pada setiap kelasnya tentu akan memudahkan sistem dalam melakukan klasifikasi karena citra latih yang ada memiliki semua bentuk kemungkinan citra yang memiliki kesamaan nilai ekstraksi fitur dengan citra kueri. Sedangkan dataset yang memiliki komposisi data citra yang beragam akan menyulitkan sistem dalam melakukan klasifikasi karena citra latih yang tersedia terbatas, sehingga ketika citra kueri diinputkan dalam sistem, sistem tidak bisa konsisten memberikan perbandingan nilai ekstraksi fitur karena citra yang latih yang tersedia tidak memiliki semua bentuk kemungkinan citra yang mempunyai kesamaan nilai ekstraksi fitur dengan citra kueri.



Gambar 6.1 Grafik hasil pengujian nilai  $k$

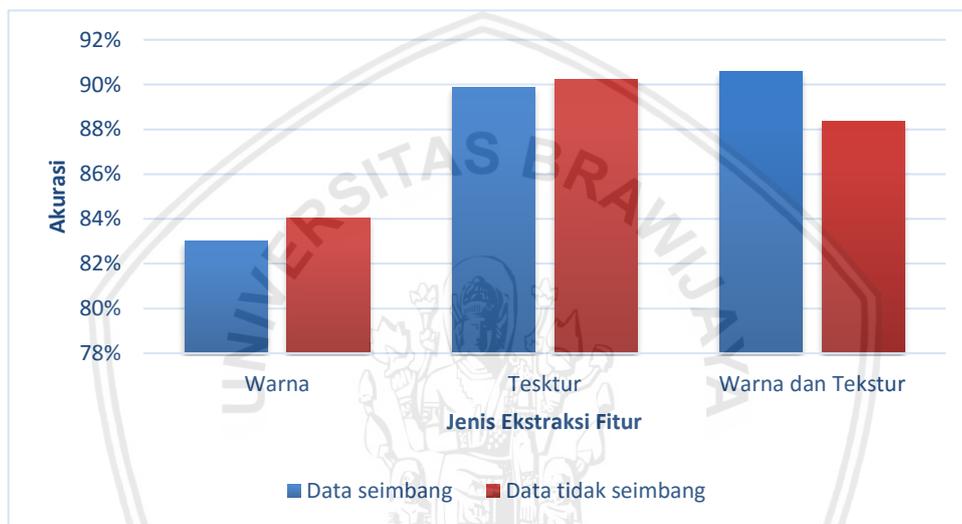
### 6.1.2 Pengujian Ekstraksi Fitur

Pengujian ekstraksi fitur bertujuan untuk mengetahui pengaruh penggunaan fitur ekstraksi terhadap akurasi hasil klasifikasi citra jenis makanan. Pengujian ini dilakukan pada proses pengambilan nilai ekstraksi fitur baik *color channel* atau GLCM. Seperti yang kita ketahui bahwa ekstraksi fitur merupakan langkah penting dalam proses pengklasifikasian citra jenis makanan. Hal ini disebabkan karena dengan melihat nilai-nilai ekstraksi fitur itulah jenis makanan satu dengan makanan yang lain dapat dibedakan. Setiap metode ekstraksi fitur memiliki perhitungan tersendiri untuk menghasilkan nilai ekstraksi fiturnya. Dengan demikian, pengujian ini diharapkan dapat memberikan informasi terkait jenis metode ekstraksi yang menghasilkan akurasi terbaik dalam mengklasifikasikan jenis makanan. Selain mengetahui pengaruh metode ekstraksi yang digunakan terhadap hasil akurasi, pengujian ini juga bertujuan untuk mengetahui jenis data yang memiliki akurasi tertinggi dalam mengklasifikasikan jenis makanan. Jenis data yang dimaksud adalah jenis data seimbang dan data

tidak seimbang. Berdasarkan Tabel 6.4 berikut hasil pengujian ekstraksi fitur menunjukkan bahwa data citra dengan ekstraksi fitur kombinasi pada data seimbang memiliki akurasi tertinggi yaitu 90,58% dan ekstraksi fitur tekstur memiliki nilai akurasi tertinggi pada data tidak seimbang yaitu 90,22%.

**Tabel 6.4 Pengujian ekstraksi fitur**

Ekstraksi Fitur	Jumlah Sub Fitur	Akurasi	
		Data seimbang	Data tidak seimbang
Warna	9	83,02%	84,04%
Tekstur	20	89,87%	90,22%
Kombinasi	29	90,58%	88,35%



**Gambar 6.2 Grafik hasil pengujian ekstraksi fitur**

Berdasarkan hasil pengujian yang ditunjukkan pada grafik Gambar 6.3, dapat diketahui bahwa dengan menggunakan data yang seimbang akurasi pada ekstraksi fitur kombinasi merupakan akurasi tertinggi dibandingkan dengan akurasi pada ekstraksi fitur warna dan tekstur. Hal ini disebabkan karena faktor komposisi data. Faktor komposisi data ini berhubungan dengan banyaknya data dalam kelas pada suatu fitur. Seperti yang sudah dijelaskan pada hasil analisis pengujian sebelumnya, yaitu pengujian nilai  $k$ , bahwa komposisi data yang tidak konsisten akan memengaruhi hasil akurasi klasifikasi citra jenis makanan dimana sistem tidak dapat menemukan nilai fitur ekstraksi yang similar dengan nilai ekstraksi pada citra kueri. Fenomena tersebut sering terjadi pada data yang memiliki komposisi seragam (data seimbang). Selain itu, jumlah sub fitur juga memengaruhi hasil akurasi karena jika dilihat pada grafik pada Tabel 6.4, semakin banyak jumlah sub fitur maka akan semakin tinggi akurasi fitur tersebut, sehingga jumlah sub fitur yang banyak akan mendominasi akurasi sistem.

Pengujian ini menunjukkan bahwa fitur terbaik yang dapat digunakan sebagai fitur untuk mengekstraksi citra jenis makanan pada penelitian ini adalah fitur tekstur. Hal ini disebabkan karena hasil akurasi dari penggunaan kedua jenis kategori data yaitu data seimbang dan data tidak seimbang memiliki selisih

akurasi yang paling kecil yaitu 0,35% dibandingkan dengan selisih akurasi pada fitur warna dan kombinasi yang masing-masing sebesar 1,02% dan 2.23%.

### 6.1.3 Pengujian Perhitungan Jarak Tetangga Terdekat

Pengujian ini merupakan pengujian yang dilakukan dengan cara membandingkan penggunaan jenis perhitungan jarak tetangga terdekat. Seperti yang kita ketahui bahwa terdapat 2 jenis perhitungan jarak untuk mengetahui tetangga terdekat dengan nilai jarak terkecil yang diurutkan berdasarkan nilai  $k$ . Kedua metode perhitungan jarak ini akan menghasilkan nilai jarak yang berbeda. Perbedaan nilai jarak yang dihasilkan tentu akan memengaruhi hasil akurasi pengklasifikasian citra jenis makanan. Hal tersebutlah yang menjadi dasar pengujian ini. Dengan demikian, pengujian ini diharapkan dapat memberikan jawaban terkait metode perhitungan jarak yang memiliki nilai akurasi tertinggi untuk dapat mengklasifikasikan citra jenis makanan. Pengujian ini menggunakan beberapa variabel yang di atur sebagai *default* seperti nilai  $k= 3, k= 5, k= 7, k= 9, k= 11, k= 13$ , dan  $k= 15$ , penggunaan data seimbang dan dan tidak seimbang, serta kombinasi ekstraksi fitur. Tabel 6.5 dan Tabel 6.6 berikut menunjukkan hasil pengujian perhitungan jarak tetangga terdekat.

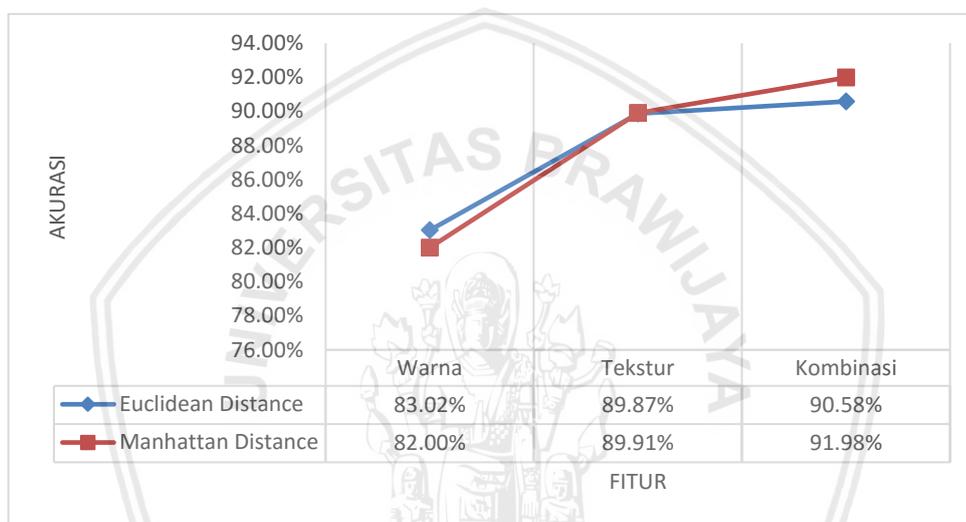
**Tabel 6.5 Pengujian perhitungan jarak tetangga terdekat**

Metode jarak terdekat	Akurasi					
	Data seimbang			Data tidak seimbang		
	Warna	Tekstur	Kombinasi	Warna	Tekstur	Kombinasi
Euclidean Distance	83,02%	89,87%	90,58%	84,04%	90,22%	88,35%
Manhattan Distance	82,00%	89,91%	91,82%	82,98%	90,44%	90,89%

Berdasarkan Tabel 6.5 tersebut di atas dapat diketahui bahwa hasil akurasi tertinggi klasifikasi citra jenis makanan dengan menggunakan metode perhitungan jarak *Euclidean* lebih kecil dibandingkan dengan menggunakan perhitungan jarak *manhattan*. Pada umumnya dalam banyak penelitian, akurasi klasifikasi dengan menggunakan *Euclidean distance* memiliki akurasi yang lebih tinggi dibandingkan dengan menggunakan *manhattan distance*. Namun, hasil pengujian ini cukup mengejutkan karena perbandingan akurasi keduanya berbeda dengan perbandingan akurasi dengan penelitian lain dengan studi kasus yang sama menggunakan metode K-NN. Pernyataan tersebut dibuktikan dengan melihat perbandingan persentasi akurasi tertinggi antara metode *Euclidean distance* dan *manhattan distance* pada masing-masing jenis komposisi dataset. Pada metode *Euclidean distance*, kedua fitur kombinasi memiliki akurasi tertinggi yaitu sebesar 90,58% pada data seimbang, sedangkan pada data tidak seimbang fitur tekstur memiliki akurasi tertinggi yaitu sebesar 90,22%. Pada metode *manhattan distance*, fitur kombinasi memiliki akurasi tertinggi pada kedua jenis kategori data masing-masing sebesar 91,82% pada data seimbang dan 90,89% pada data tidak seimbang. Hal ini tentu mematahkan sebuah yang menyatakan

bahwa jarak *Euclidean* merupakan metode pengukur jarak dalam arti yang sebenarnya. Pernyataan yang diungkapkan oleh Etienne Folio dalam jurnalnya tentang *Distance Transform* tahun 2008 tersebut rupanya belum dapat dipastikan benar untuk beberapa studi kasus seperti pada penelitian ini khususnya jika dibandingkan dengan menggunakan metode *manhattan distance*. Sistem pengenalan jenis makanan berbasis citra dengan perbedaan perhitungan jarak dapat menghasilkan output yang masih sangat kondisional dari segi akurasi.

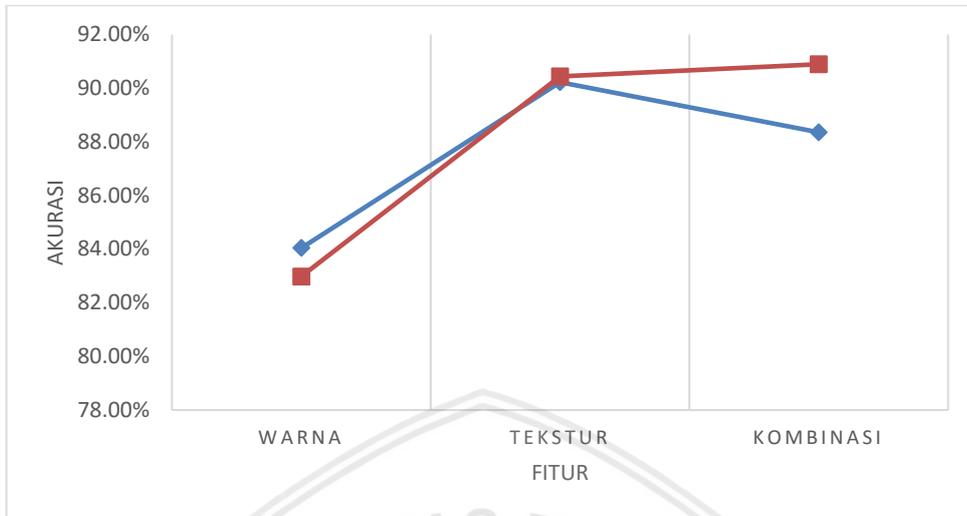
Di samping itu, penggunaan perhitungan jarak baik dengan *Euclidean distance* atau *manhattan distance*, keduanya menghasilkan akurasi tertinggi pada fitur yang sama yaitu fitur kombinasi. Keduanya memiliki akurasi masing-masing sebesar 91,82% dan 90,58%. Fenomena tersebut dapat dilihat pada grafik Gambar 6.3 berikut ini.



**Gambar 6.3** Grafik hasil pengujian perhitungan jarak pada data seimbang

Data tersebut menunjukkan bahwa fitur kombinasi merupakan fitur yang optimal untuk digunakan dalam mengidentifikasi citra jenis makan dengan komposisi data yang seimbang. Khusus untuk data seimbang, jika ingin menggunakan ekstraksi fitur warna saja dalam penelitian ini, maka hasil akurasi yang optimal akan didapat jika menggunakan *Euclidean distance*. Jika menggunakan ekstraksi fitur tekstur, maka *manhattan distance* akan menghasilkan akurasi yang optimal. Selain itu, jumlah sub fitur merupakan salah satu faktor yang mempengaruhi akurasi pada kedua jenis perhitungan jarak karena semakin besar jumlah sub fitur, maka akurasi yang dihasilkan akan semakin besar. Lain halnya dengan data tidak seimbang, akurasi terbaik sistem dalam mengklasifikasikan citra jenis makanan adalah dengan menggunakan ekstraksi fitur tekstur dengan akurasi sebesar 90,22% untuk metode *Euclidean distance* serta menggunakan ekstraksi fitur tekstur dengan akurasi sebesar 90,89% untuk metode *manhattan distance*. Dengan kata lain khusus untuk data tidak seimbang, penggunaan metode *manhattan distance* akan sangat akurat jika ekstraksi fitur yang digunakan adalah fitur kombinasi, sedangkan metode *Euclidean distance* akan sangat akurat jika ekstraksi fitur yang digunakan adalah

fitur tesktur. Berikut grafik pada Gambar 6.4 menunjukkan perbedaan penjelasan tersebut di atas.



Gambar 6.4 Grafik hasil pengujian perhitungan jarak pada data tidak seimbang



## BAB 7 PENUTUP

### 7.1 Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan, didapatkan kesimpulan sebagai berikut:

1. Dari hasil pengujian yang dilakukan, tingkat akurasi sistem pengenalan citra jenis makanan yang menerapkan ekstraksi fitur *color channel* dan *gray level co-occurrence matrix* dengan metode *k-nearest neighbor* ditentukan oleh 3 faktor yaitu:
  - Nilai *k*, penggunaan *k=3* merupakan nilai *k* yang optimal untuk digunakan pada jenis data seimbang dan tidak seimbang dengan akurasi masing-masing sebesar 90,58% dan 88,35%.
  - Ekstraksi fitur, penggunaan ekstraksi fitur tekstur akan menghasilkan akurasi yang optimal pada data seimbang maupun data tidak seimbang karena selisih akurasi yang terkecil yaitu sebesar 0,35%
  - Perhitungan jarak, penggunaan *Euclidean distance* akan menghasilkan akurasi optimal untuk fitur kombinasi pada data seimbang dengan akurasi sebesar 90,58%. Jika menggunakan *manhattan distance*, fitur kombinasi pada data seimbang menghasilkan akurasi optimal sebesar 91,82%.
2. Ekstraksi fitur *color channel* dan *gray level CO-OCCURENCE MATRIX* dengan metode *k-nearest neighbor* terhadap hasil akurasi dipengaruhi oleh beberapa faktor yaitu:
  - Fitur warna akan menghasilkan akurasi optimal pada data tidak seimbang dengan akurasi sebesar 84,04%.
  - Fitur tesktur akan menghasilkan akurasi optimal pada data tidak seimbang dengan akurasi sebesar 90,22%.
  - Fitur kombinasi (tekstur dan warna) akan menghasilkan akurasi optimal pada data seimbang dengan akurasi sebesar 90,58%.

### 7.2 Saran

Penelitian yang telah dilakukan masih jauh dari kata sempurna dan perlu dilakukan banyak pengembangan. Adapun beberapa saran untuk penelitian selanjutnya yaitu:

1. Perlu dilakukan penelitian lebih lanjut terkait penggunaan metode ekstraksi warna dengan channel lain misalnya HSV, YUV, LBP, dan sebagainya .
2. Perlu dilakukan penambahan metode seleksi fitur untuk dapat mengetahui fitur mana yang paling berpengaruh terhadap hasil klasifikasi sehingga dapat meningkatkan akurasi pada sistem pengenalan citra jenis makanan ini.

## DAFTAR REFERENSI

- Afianto, T. & Amalia, F., 2016. *Pengaruh Komponen Krominan Pada Ruang Warna HSV, YCBCR, dan CIELAB Untuk Deteksi Kulit Menggunakan Klasifikasi KNN*. Yogyakarta, ResearchGate.
- Agaputra, M. D., Wardani, K. R. R. & Siswanto, E., 2013. Pencarian Citra Digital Berbasis Konten dengan Ekstraksi Fitur HSV, ACD, dan GLCM. *Jurnal Telematika*, Volume VIII, pp. 10-11.
- Anami, B. S. & Burkpalli, V. C., 2009. Texture Based Identification and Classification of Bulk Sugary Food Objects. *Graphics, Vision, and Image Processing Journal*, 9(4), pp. 9-12.
- Arivazhagan, S., Shebiah, R. N., Nidhyandhan, S. & Ganesan, L., 2010. Fruit Recognition Using Color and Texture Features. *Journal of Emerging Trends in Computing and Information Science*, Volume 1, pp. 90 - 93.
- Bhargavi, K. & Jyothi, S., 2014. A Survey on Threshold Based Segmentation Technique in Image Processing. *International Journal Of Innovative Research & Development*, III(12), pp. 234-235.
- Bozkurt, F., Yağanoğlu, . M. & Günay, F. B., 2015. Effective Gaussian Blurring Process on Graphics. *International Journal of Machine Learning and Computing*, V(1), p. 58.
- Dhanachandra, N., Manglem, K. & Chanu, Y. J., 2015. *Image Segmentation using K-means Clustering Algorithm and*. National Institute of Technology, Manipur, India., *Procedia Computer Science*.
- Farinella, G. M., Battiato, S. & Moltisanti, M., 2014. *Classifying food images represented as Bag of Textons*. University of Catania, ResearchGate, pp. 1-6.
- Gonzales, R. C. & Woods, R. E., 2002. *Digital Image Processing*. 2nd penyunt. New Jersey: Prentice Hall.
- He, Y. et al., 2014. Analysis of Food Images : Feature and Classification. *Manuscript*, pp. 1-4.
- He, Y. et al., 2014. *Analysis of Food Images: Features And Classification*. Purdue University, Graphic Era University, University of Hawaii Cancer Center, HHS Public Access.
- Hoashi, H., Joutou, T. & Yanai, K., 2016. *Image Recognition of 85 Food Categories by Feature Fusion*. Tokyo, Department of Computer Science, The University of Electro-Communication.
- Kadir, A. & Susanto, A., 2013. *Teori dan Aplikasi Pengolahan Citra*. 1st penyunt. Yogyakarta: Andi.
- Karegowda, D. G., S, S. & K.R, P., 2015. Automated Detection and Classification of Images Using Color and Texture Features-A Survey. *International Journal of*

*Innovative Research in Computer and Communication Engineering*, Volume 3, pp. 249-251.

Kodituwaku, S. & Selvarajah, S., 2010. Comparison of Color Features for Image Retrieval. *Indian Journal of Computer Science and Engineering*, 1(3), p. 208.

Loesdau, M., Chabrier, S. & Gabillon, A., 2014. *Hue and Saturation in the RGB Color Space*. Tahiti, ResearchGate.

Lu, J., s.l. *Classification Accuracy and Model Selection in k-Nearest Neighbors Classifiers for Data Driven Learning*, Henan, China: Xinxiang University.

Lu, S., Lin, S. & Wang, B., 2012. Recognition and Classification of Fast Food Images. pp. 43-45.

Ma, C. M., Yang, W. S. & Cheng, B. W., 2014. How the Parameters of K-nearest Neighbor Algorithm Impact on the Best Classification Accuracy: In Case of Parkinson Dataset. *Journal of Applied Sciences*, XIV(2), pp. 171-176.

N., S. & C., K., 2014. A Case Study on Mathematical Morphology Segmentation for MRI Brain Image. *International Journal of Computer Science and Information Technologies*, V(4), pp. 5337-5338.

N. & Fernando, Y., 2017. *Klasifikasi Jenis Daging Berdasarkan Analisis Citra Tesktur Gray Level Co-Occurance Matrices (GLCM) dan Warna*. Jakarta, Fakultas Teknik Universitas Muhammadiyah, p. 2.

Patil, N. K., Malemath, V. S. & Yadahalli, R. M., 2011. Color and Texture Based Identification and Classification of Food Grains Using Different Color Models and Haralick Features. *International Journal on COmputer Science and Engineering (UCSE)*, Volume 3, pp. 3669-3677.

Ponnoli, K. & Selvamuthukumar, S., 2014. Analysis of Face Recognition using Manhattan Distance Algorithm with Image Segmentation. *International Journal of Computer Science and Mobile Computing*, III(7), p. 19.

Priandana, K., S., A. Z. & S., 2016. Mobile Munsell Soil Color Chart Berbasis Android Menggunakan Histogram Ruang Citra HVC dengan Klasifikasi KNN. *Jurnal Ilmu Komputer Agri-Informatika*, III(2), pp. 95-96.

Qur'ania, A., Karlitasar, L. & Maryana, S., 2012. Analisis Tekstur dan Ekstraksi Fitur Warna Untuk Klasifikasi Apel Berbasis Citra. *Lokakarya Komputasi dalam Sains dan Teknologi Nuklir*, 10 October, pp. 296-304.

Sari, Y. A., Dewi, R. K. & Fatichah, C., 2014. Seleksi Fitur Menggunakan Ekstraksi Fitur Bentuk, Warna, dan Tekstur Dalam Sistem Temu Kembali Citra Daun. *Juti*, 12(1), pp. 1-8.

Shaykh, R., 2018. *Towards Data Science*. [Online] Available at: <https://towardsdatascience.com> [Diakses 17 December 2018].

Syafitri, N., 2008. *Perbandingan Metode K-Nearest Neighbor (K-NN) dan Nearest Cluster Classifier (NCC) Untuk Klasifikasi Kualitas Batik Tulis*. Yogyakarta, UGM Library.

Wibisono, A., 2009. Hubungan Antara Penglihatan, Pencahayaan, dan Persepsi Manusia Dalam Desain Interior. *AMBIANCE*, pp. 89-90.

Wibowo, A., 2017. *10 Fold-Cross Validation*. [Online] Available at: <https://mti.binus.ac.id> [Diakses 16 January 2019].

Zhou, H., Wu, J. & Zhang, J., 2010. *Digital Image Processing: Part I*. 1st penyunt. s.l.:Ventus Publishing Aps.

