

**Pengenalan Kematangan Buah Nanas
Menggunakan Metode *Gray Level
Co-Occurrence Matrix* (GLCM)
dan *K-Nearest Neighbor* (K-NN)**

Oleh

MOHAMAD RIZAL KASIM

T3116067

SKRIPSI

**Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar sarjana**



**PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2021/2022**

HALAMAN JUDUL

**Pengenalan Kematangan Buah Nanas
Menggunakan Metode *Gray Level
Co-Occurrence Matrix* (GLCM)
Dan *K-Nearest Neighbor* (K-NN)**

Oleh

MOHAMAD RIZAL KASIM

T3116067

SKRIPSI

**Untuk memenuhi salah satu syarat ujian
Guna memperoleh gelar sarjana**



**PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2021/2022**

PENGESAHAN SKRIPSI

Pengenalan Kematangan Buah Nanas Menggunakan Metode *GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM)* Dan *K-NEAREST NEIGHBOR (K-NN)*

Oleh

MOHAMAD RIZAL KASIM

T3116067

SKRIPSI

Untuk memenuhi salah satu syarat ujian
Guna memperoleh gelar sarjana
Program Studi Teknik Informatika
Ini telah disetujui oleh Tim Pembimbing
Gorontalo, Juni 2021

Pembimbing Utama



Husdi, M.Kom

NIDN. 0907108701

Pembimbing Pendamping



Yusrianto Malago, M.Kom

NIDN. 0909108901

PERSETUJUAN SKRIPSI

Pengenalan Kematangan Buah Nanas Menggunakan Metode *Gray Level Co-Occurrence Matrix (GLCM)* dan *K-Nearest Neighbor (K-NN)*

Oleh

MOHAMAD RIZAL KASIM

T3116067

Diperiksa oleh panitia ujian Strata Satu (S1)

Universitas Ichsan Gorontalo

Gorontalo, Desember 2021

1. Penguji I
Haditsa Annur, M.Kom
2. Penguji II
Sudirman S. Panna, M.Kom
3. Penguji III
Andi Kamaruddin, M.Kom
4. Pembimbing I
Husdi, M.Kom
5. Pembimbing II
Yusrianto Malago, M.Kom

.....
.....
.....
.....
.....



Dekan Fakultas

Jerry Karim, M.Kom

NIDN. 0918077302



Ketua Jurusan

Sudirman S. Panna, M.Kom

NIDN. 0908017702

PERNYATAAN SKRIPSI

Dengan ini saya menyatakan bahwa :

1. Karya tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorontalo maupun di perguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari Tim Pembimbing.
3. Dalam Karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah dipublikasi orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pula dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, Juni 2021

buat Pernyataan



Mohamad Rizal Kasim

ABSTRACT

MOHAMAD RIZAL KASIM. T3116067. RECOGNITION OF PINEAPPLE RIPENESS USING GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM) AND K-NEAREST NEIGHBOR (K-NN) METHODS

Ananas comosus or the popularly known as pineapple is a fruit that grows in the tropics and subtropics. Pineapple is also one of the favorite fruits consumed by the community, whether consumed as a fruit or in processed form. To find the perfect level of pineapple ripeness, it is generally selected manually. However, it becomes less efficient if the pineapples are selected in large quantities. Therefore, the image processing techniques are required to determine the maturity level of pineapples automatically using feature extraction of Gray Level Co-occurrence Matrix and K-Nearest Neighbor as the classification models which have not been tested before but are proven to be able and reliable in solving problems such as determining the maturity level of pineapple. The findings of the study indicate that the accuracy of the K-NN model for determining the ripeness level of pineapple is 83% with values of $K=9$ and $K=7$ with GLCM direction 0^0 and distance 1. This study employs 30 testing data and 209 training data, namely ripe and raw data with 115 mature data respectively, and 94 raw data. Thus, the application of the Gray Level Co-occurrence Matrix and K-Nearest Neighbor methods successfully get a high accuracy value of 83% using the Confusion Matrix test.

Keywords: pineapple, recognition, Gray Level Co-Occurrence Matrix, K-Nearest Neighbor



ABSTRAK

MOHAMAD RIZAL KASIM. T3116067. PENGENALAN KEMATANGAN BUAH NANAS MENGGUNAKAN METODE *GRAY LEVEL CO-OCCURRENCE MATRIX* (GLCM) DAN *K-NEAREST NEIGHBOR* (K-NN).

Ananas comosus atau yang kita kenal dengan buah nanas merupakan buah yang tumbuh di daerah tropis dan subtropis. Buah nanas juga merupakan salah satu buah favorit yang dikonsumsi masyarakat, baik dikonsumsi secara buah ataupun dalam bentuk olahan. Untuk menemukan tingkat kematangan buah nanas yang sempurna umumnya dipilih secara manual tetapi, hal ini menjadi kurang efisien apabila buah nanas yang diseleksi dalam jumlah yang banyak. Maka dari itu dibutuhkan teknik pengolahan citra dalam menentukan tingkat kematangan buah nanas secara otomatis menggunakan ekstraksi fitur *Gray Level Co-occurrence Matrix* dan *K-Nearest Neighbor* sebagai model Klasifikasinya yang belum pernah di uji coba sebelumnya namun terbukti dapat dan handal dalam menyelesaikan masalah seperti penentuan tingkat kematangan buah nanas. Hasil penelitian menunjukkan bahwa akurasi model K-NN untuk penentuan tingkat kematangan buah nanas sebesar 83% pada nilai $K=9$ dan $K=7$ dengan GLCM arah 0^0 dan jarak 1. Penelitian ini menggunakan 30 data testing dan 209 data training, yaitu kategori matang dan mentah dengan masing-masing 115 data Matang, dan 94 data Mentah. Dengan demikian, Penerapan metode *Gray Level Co-occurrence Matrix* dan *K-Nearest Neighbor* pada masalah diatas berhasil mendapatkan nilai akurasi yang tinggi sebesar 83% dengan menggunakan pengujian *Confusion Matrix*.



Kata Kunci : Nanas, Pengenalan, *Gray Level Co-Occurrence Matrix*, *K-Nearest Neighbor*.

KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan usulan penelitian proposal ini dengan judul : “Pengenalan Kematangan Buah Nanas Menggunakan Metode *Gray Level Co-Occurrence Matrix* (GLCM) dan *K-Nearest Neighbor* (K-NN)”, untuk memenuhi salah satu syarat penyusunan Skripsi Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penulis menyadari sepenuhnya bahwa usulan penelitian ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materil. Untuk itu, dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan, banyak terima kasih dan penghargaan yang setinggi-tingginya kepada :

1. Bapak Muhammad Ichsan Gaffar, SE, M.Ak, selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo;
2. Bapak Dr. Abdul Gaffar La Tjokke, M.Si, selaku Rektor Universitas Ichsan Gorontalo;
3. Bapak Jorry Karim, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
4. Bapak Sudirman Melangi, M.Kom, selaku Wakil Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
5. Ibu Irma Surya Kumala Idris, M.Kom, selaku Wakil Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
6. Bapak Sudirman S. Panna, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
7. Husdi, M.Kom, selaku Pembimbing I;
8. Yusrianto Malago, M.Kom, selaku Pembimbing II;
9. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan telah mengajarkan berbagai disiplin ilmu kepada penulis;

10. Kedua Orang Tua saya tercinta, atas segala kasih sayang, jerih payah, dan doa restunya dalam membesarkan dan mendidik penulis;
11. Bapak Rifai Mokoginta selaku Sangadi Lobong (Kepala desa lobong) Bersama istri yang telah membantu saya dalam proses pengumpulan data nanas di desa lobong.
12. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan dan dukungan moril yang sangat besar kepada penulis;
13. Kepada semua pihak yang ikut membantu dalam penyelesaian skripsi ini tak sempat penulis sebutkan satu-persatu;

Semoga Allah SWT, melimpahkan balasan atas jasa-jasa mereka kepada kami. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan adanya kritik dan saran yang konstruktif. Akhirnya penulis berharap semoga hasil yang telah dicapai ini dapat bermanfaat bagi kita semua, Aamiin.

Gorontalo, Juni 2021

Mohamad Rizal Kasim

DAFTAR ISI

HALAMAN JUDUL.....	i
PENGESAHAN SKRIPSI	ii
PERSETUJUAN SKRIPSI	iii
PERNYATAAN SKRIPSI.....	iv
<i>ABSTRACT</i>	v
ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN	1
1. 1. Latar Belakang	1
1. 2. Identifikasi Masalah	3
1. 3. Rumusan Masalah	3
1. 4. Tujuan Penelitian	3
1. 5. Manfaat Penelitian	3
1. 5. 1. Manfaat Teoritis.....	3
1. 5. 2. Manfaat Praktis.....	3
BAB II LANDASAN TEORI	4
2. 1. Tinjauan Studi	4
2. 2. Tinjauan Pustaka	6
2. 2. 1. Buah Nanas	6

2. 2. 2. Kematangan Buah	6
2. 2. 3. Citra (<i>image</i>)	7
2. 2. 4. Metode <i>Gray Level Co-Occurance Matrix</i> (GLCM)....	7
2. 2. 5. Metode <i>K-Nearest Neighbor</i> (K-NN)	10
2. 2. 6. Ekstrasi Fitur	14
2. 2. 7. Perangkat Lunak Pendukung	14
2. 2. 8. Evaluasi Model.....	14
2. 2. 9. <i>Confusion Matrix</i>	15
2. 3. Kerangka Pikir	17
BAB III METODE PENELITIAN.....	18
3. 1. Jenis, Metode, Subjek, Objek, Waktu dan Lokasi Penelitian	18
3. 2. Pengumpulan Data	18
3. 3. Pemodelan	19
3. 3. 1. Pra Pengolahan Data.....	19
3. 3. 2. Ekstraksi Ciri.....	20
3. 3. 3. Data Training.....	20
3. 3. 4. Training Menggunakan K-NN	20
3. 3. 5. Model	20
3. 3. 6. Data Testing	20
3. 3. 7. Hasil Klasifikasi.....	21
3. 3. 8. Evaluasi	21
BAB IV HASIL PENELITIAN	22
4. 1 Hasil Pengumpulan Data	22
4. 2 Hasil Pemodelan	22
4.2.1 Pengambilan Citra Buah Nanas	22

4.2.2	Pra-Pengolahan	23
4.2.3	Ekstraksi Fitur	25
4.2.4	Klasifikasi.....	28
4.2.5	Evaluasi Model	34
4.2.6	<i>System Requirements</i>	41
BAB V	PEMBAHASAN	42
5.1	Pembahasan Model	42
5.2	Pembahasan Sistem	44
BAB VI	PENUTUP	46
6.1	Kesimpulan	46
6.2	Saran.....	46
DAFTAR PUSTAKA		xvi
LAMPIRAN		

DAFTAR GAMBAR

GAMBAR 2. 1	GAMBAR BUAH NANAS	6
GAMBAR 2. 2	DIAGRAM HASIL PENGUJIAN.....	13
GAMBAR 2. 3	KERANGKA PIKIR.....	17
GAMBAR 3. 1	PEMODELAN.....	19
GAMBAR 4. 1	CONTOH ILUSTRASI PENGAMBILAN PIXEL PADA CITRA GRAYSCALE	24
GAMBAR 5. 1	TAMPILAN MENU UTAMA	44
GAMBAR 5. 2	TAMPILAN PROSES LOAD DATA TRAINING.....	44
GAMBAR 5. 3	TAMPILAN PROSES LOAD DATA TESTING	45
GAMBAR 5. 4	TAMPILAN HASIL PENGENALAN	45

DAFTAR TABEL

TABEL 2. 1 FORMAT TINJAUAN STUDI.....	4
TABEL 2. 2 SAMPLE DATA TRAINING.....	11
TABEL 2. 3 SAMPLE DATA TESTING.....	11
TABEL 2. 4 SAMPEL HASIL PENGUJIAN.....	12
TABEL 2. 5 HASIL PENGUJIAN	13
TABEL 2. 6 PERANGKAT LUNAK PENDUKUNG	14
TABEL 2. 7 CONFUSION MATRIX	15
TABEL 4. 1 DATA GAMBAR BUAH NANAS	22
TABEL 4. 2 CITRA GRAYSCALE 6X6	24
TABEL 4. 3 PERHITUNGAN DENGAN LANGKAH DAN RUMUS	24
TABEL 4. 4 CITRA ASLI 6X6	25
TABEL 4. 5 DATA CITRA 6X6 SETELAH DILAKUKAN HISTOGRAM EQUALIZATION.....	25
TABEL 4. 6 NILAI DARI SUATU CITRA	26
TABEL 4. 7 MATRIX GLCM.....	26
TABEL 4. 8 MATRIX NORMALISASI	26
TABEL 4. 9 HASIL EKSTRAKSI FITUR.....	29
TABEL 4. 10 PERHITUNGAN JARAK EUCLIDEAN DISTANCE	30
TABEL 4. 11 HASIL PENGURUTAN DATA	30
TABEL 4. 12 NILAI K=1	31
TABEL 4. 13 KELAS DATA UJI DARI NILAI K=1.....	31
TABEL 4. 14 NILAI K=3	31
TABEL 4. 15 KELAS DATA UJI DARI NILAI K=3.....	31
TABEL 4. 16 NILAI K=5	32
TABEL 4. 17 KELAS DATA UJI DARI NILAI K=5.....	32
TABEL 4. 18 NILAI K=7	32
TABEL 4. 19 KELAS DATA UJI DARI NILAI K=7.....	32
TABEL 4. 20 NILAI K=9	33

TABEL 4. 21 KELAS DATA UJ DARI NILAI $K=9$	33
TABEL 4. 22 HASIL PERHITUNGAN EUCLIDEAN DISTANCE UNTUK DATA TESTING.....	33
TABEL 4. 23 HASIL DATA TESTING DARI SALAH SATU NILAI K ($K=1$)	34
TABEL 4. 24 HASIL DATA UJI $K=1$	36
TABEL 4. 25 HASIL DATA UJI $K=3$	37
TABEL 4. 26 HASIL DATA UJI $K=5$	38
TABEL 4. 27 HASIL DATA UJI $K=7$	39
TABEL 4. 28 HASIL DATA UJI $K=9$	40
TABEL 5. 1 HASIL PEMBAHASAN MODEL.....	43

DAFTAR LAMPIRAN

LAMPIRAN 1. 1 KODE PROGRAM

LAMPIRAN 1. 2 RIWAYAT HIDUP PENELITI

LAMPIRAN 1. 3 SURAT BALASAN PENELITIAN

LAMPIRAN 1. 4 SURAT REKOMENDASI BEBAS PLAGIASI

BAB I

PENDAHULUAN

1. 1. Latar Belakang

Ananas comosus atau yang kita kenal dengan buah nanas merupakan buah yang tumbuh di daerah tropis dan subtropis. Buah nanas juga merupakan salah satu buah favorit yang dikonsumsi masyarakat, baik dikonsumsi secara buah ataupun dalam bentuk olahan. Hal lain yang perlu diketahui dari nanas adalah selain rasanya yang enak buah nanas memiliki banyak sekali kandungan seperti Kalori, Protein, dan Karbohidrat serta mengandung banyak vitamin. Buah nanas umumnya mencapai tingkat kematangan selama 120 – 170 hari dari mulai berbunga [1].

Pengolahan citra merupakan perkembangan teknologi dimana mesin (computer) diinginkan dapat mengenali citra seperti layaknya penglihatan manusia[2]. Pengolahan citra adalah metode atau teknik yang dapat digunakan untuk memproses citra atau gambar dengan cara memanipulasinya menjadi data citra yang diinginkan untuk memperoleh informasi tertentu [3].

Dalam penelitian ini buah nanas yang dipilih secara umum akan digunakan sebagai objek citra buah yang akan dikenali dalam tiga kategori yaitu mentah, dan matang, dengan memprediksi kematangan menggunakan fitur warna kulit nanas. Untuk menemukan tingkat kematangan buah nanas yang sempurna umumnya dipilih secara manual tetapi, hal ini menjadi kurang efisien apabila buah nanas yang diseleksi dalam jumlah yang banyak. Sehingga, dibutuhkan metode otomatis yang dapat meningkatkan akurasi dengan penilaian yang konsisten dalam klasifikasi tingkat kematangan buah nanas berdasarkan fitur warna kulit. Metode yang akan digunakan pada pengenalan kematangan buah nanas adalah *Gray Level Co-occurrence Matrix* (GLCM) dan *K-Nearest Neighbor* (K-NN). GLCM adalah suatu metode yang digunakan untuk analisis tekstur/ekstraksi ciri. GLCM merupakan suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra [4]. Kelebihan

dari metode ini misalnya jarak antara dua obyek tidak dipengaruhi oleh penambahan obyek baru ke dalam analisis [2].

Sedangkan K-NN (*K-Nearest Neighbor*) merupakan salah satu metode yang digunakan dalam pengklasifikasian. K-nn memiliki prinsip mencari data terdekat antara data yang akan dievaluasi dalam data pelatihan. Kelebihan dari metode ini yaitu jika training data cukup besar maka akan menghasilkan data yang akurat dan efektif. Tetapi biaya yang digunakan cukup besar karena, pada keseluruhan data dibutuhkan perhitungan *Query instance* untuk perhitungan jaraknya [5].

Beberapa hasil penelitian sebelumnya yang menggunakan pengolahan citra seperti yang dilakukan oleh, Restu Widodo, Agus Wahyu Widodo dan Arry Supriyanto tentang Pemanfaatan Ciri *Gray Level Co-Occurrence Matrix* (GLCM) Citra Buah Jeruk Keprok (*Citrus reticulata Blanco*) untuk Klasifikasi Mutu Dari hasil pengujian yang dilakukan pada aplikasi ini, Klasifikasi mutu kedalam 3 kelas yaitu, Grade Super, Grade A dan Grade B. Penelitian ini menghasilkan tingkat akurasi terbaik sebesar 82.5% dengan jumlah data latih sebanyak 20, nilai $distance=2$ pada arah GLCM 45°. Ada juga penelitian yang dilakukan oleh Achdiyat Risvana, Prajanto Wahyu Adi, M.Kom, dan Setia Astuti, S.Si, M.Kom tentang Klasifikasi Tingkat Kematangan Buah Jambu Biji Merah Dengan Algoritma *K-Nearest Neighbor* dari hasil pengujian yang dilakukan pada aplikasi ini mendapatkan proses klasifikasi tingkat kematangan buah jambu biji merah menggunakan $k=1$ mendapatkan akurasi sebesar 88% sedangkan menggunakan $k=3$ mendapatkan akurasi sebesar 84%.

Adapun peneliti meneliti tentang “**Pengenalan Kematangan Buah Nanas Menggunakan Metode *Gray Level Co-Occurrence Matrix* (GLCM) dan *K-Nearest Neighbor* (K-NN)**” terdapat sebuah alasan yakni penulis ingin mengetahui akurasi tingkat kematangan buah dengan menggunakan metode GLCM dan K-NN. Dengan pengolahan citra digital yang semakin berkembang, maka dalam tugas akhir ini telah dibuat suatu sistem yang dapat mengenali tingkat kematangan buah

nanas secara otomatis. Sehingga akan sangat membantu dalam menentukan tingkat kematangan buah nanas secara baik dan akurat.

1. 2. Identifikasi Masalah

Peneliti ingin melakukan percobaan tentang tingkat akurasi metode *Gray Level Co-Occurrence Matrix* (GLCM) dan *K-Nearest Neighbor* (K-NN), dalam menentukan tingkat kematangan pada buah nanas.

1. 3. Rumusan Masalah

Adapun rumusan masalah kematangan buah nanas dengan *image processing* menggunakan metode GLCM dan K-NN yaitu sebagai berikut :

1. Seberapa besar akurasi yang bisa dihasilkan dengan metode GLCM dan K-NN dalam menentukan tingkat kematangan buah nanas?
2. Bagaimana proses identifikasi tingkat kematangan buah nanas menggunakan metode GLCM dan K-NN ?

1. 4. Tujuan Penelitian

Adapun yang yang menjadi tujuan pada penelitian ini adalah:

1. Dapat melihat seberapa besar akurasi yang bisa didapatkan dengan metode GLCM dan K-NN, dalam menentukan tingkat kematangan buah nanas.
2. Dapat menerapkan teknik citra dalam dunia pertanian maupun dunia pemasaran, sehingga berguna bagi para masyarakat untuk memperoleh tingkat kematangan buah yang relatif lebih bagus.

1. 5. Manfaat Penelitian

1. 5. 1. Manfaat Teoritis

Penelitian ini diharapkan dapat menjadi referensi pada penelitian yang berhubungan dengan mengidentifikasi kematangan buah nanas dengan *Image processing* menggunakan metode GLCM dan K-NN.

1. 5. 2. Manfaat Praktis

Dapat menambah wawasan dan pengalaman tentang cara mengidentifikasi kematangan buah nanas menggunakan metode GLCM dan K-NN.

BAB II

LANDASAN TEORI

2. 1. Tinjauan Studi

Adapun penelitian yang sebelumnya dan berhubungan dengan judul peneliti untuk dijadikan di jadikan sebagai acuan atau referensi untuk menentukan metode yang akan digunakan nantinya.

Tabel 2. 1 Format Tinjauan Studi

NO	PENELITI	JUDUL/TAHUN	METODE	HASIL
1	Restu Widodo Agus Wahyu Widodo Arry Supriyanto	Pemanfaatan Ciri <i>Gray Level Co-Occurrence</i> <i>Matrix (GLCM)</i> Citra Buah Jeruk Keprok (Citrus <i>reticulata</i> Blanco) untuk Klasifikasi Mutu / 2018	Menggunakan metode <i>Gray Level Co-Occurrence Matrix (GLCM)</i>	Hasil ekstraksi ciri <i>gray level co-occurrence matrix (GLCM)</i> citra jeruk keprok untuk klasifikasi mutu dapat memberikan hasil terbaik sebesar 82.5% dengan jumlah data latih sebesar 20 dan 30 data, pada jarak ketetanggaan piksel (distance) sebesar 2 dan pada arah GLCM 45°.

NO	PENELITI	JUDUL/TAHUN	METODE	HASIL
2	Achdiyat Risvana Prajanto Wahyu Adi,M.Kom Setia Astuti,S.Si,M.Kom	Klasifikasi Tingkat Kematangan Buah Jambu Biji Merah Dengan Algoritma <i>K-Nearest Neighbor</i>	Menggunakan metode <i>K-Nearest Neighbor</i>	Hasil proses klasifikasi tingkat kematangan buah jambu biji merah menggunakan k=1 mendapatkan akurasi sebesar 88% sedangkan menggunakan k=3 mendapatkan akurasi sebesar 84%.
3	Ayu Pariyandani Diah Ayu Larasati Eka Pirdia Wanti Muhathir	Klasifikasi Citra Ikan Berformalin Menggunakan Metode K-NN dan GLCM / 2019	Menggunakan Metode <i>Gray Level Co-Occurrence Matrix</i> (GLCM) dan <i>K-Nearest Neighbor</i> (K-NN)	Ikan segar sebagai ikan segar sebesar 53% dan ikan berformalin dikenali sebagai ikan yang berformalin sebesar 92%.

2. 2. Tinjauan Pustaka

2. 2. 1. Buah Nanas

Ananas comosus atau yang kita kenal dengan buah nanas merupakan buah yang tumbuh di daerah tropis dan subtropis. Buah nanas secara umum memiliki 4 jenis varietas yakni *cayenne*, *queen*, *spanish*, *abacaxi*. Buah nanas juga merupakan salah satu buah favorit yang dikonsumsi masyarakat, baik dikonsumsi secara buah ataupun dalam bentuk olahan. Hal lain yang perlu diketahui dari nanas adalah selain rasanya yang enak buah nanas memiliki banyak sekali kandungan seperti Kalori, Protein, dan Karbohidrat serta mengandung banyak vitamin [1].



Gambar 2. 1 Gambar Buah Nanas

2. 2. 2. Kematangan Buah

Buah nanas umumnya mencapai tingkat kematangan selama 120 – 170 hari dari mulai berbunga [1].

1. Muda atau mentah memiliki warna hijau tua, belum layak dipetik,
2. Tingkat permulaan matang, memiliki warna hijau tua dengan sedikit warna kuning antara mata dibagian pangkal,
3. Matang, hampir 50% mata berubah menjadi warna orange kekuning-kuningan,

2. 2. 3. Citra (*image*)

Citra adalah representasi, imitasi, atau kemiripan dari suatu objek. Citra sebagai *Output* suatu sistem perekaman data dapat bersifat optic berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada layar televisi, atau bersifat digital yang dapat langsung disimpan pada suatu penyimpanan [6]. Citra memiliki 3 jenis yaitu [6]:

1. Citra Berwarna

Citra berwarna atau citra RGB, adalah jenis citra yang menyajikan warna dalam bentuk komponen R (merah), G (hijau), dan B (biru). Setiap komponen warna menggunakan delapan bit (nilainya berkisar antara 0 sampai dengan 255). Dengan demikian, kemungkinan warna yang dapat disajikan mencapai $255 \times 255 \times 255$ atau 16.581.375 warna.

2. Citra Keabuan

Citra ini menangani gradiasi warna hitam dan putih, yang menghasilkan efek warna abu-abu. Pada jenis ini, warna dinyatakan dengan intensitas. Dalam hal ini, intensitas berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih.

3. Citra Biner

Citra biner adalah citra yang setiap piksel dinyatakan dengan nilai dari dua kemungkinan (yaitu nilai 0 dan 1). Nilai 0 sebagai warna hitam dan nilai 1 sebagai warna putih. Citra jenis ini sering dipakai dalam pemrosesan citra, misalnya untuk kepentingan memperoleh tepi bentuk suatu objek.

2. 2. 4. Metode *Gray Level Co-Occurance Matrix* (GLCM)

Gray Level Co-Occurance Matrix atau yang disingkat GLCM adalah suatu metode yang digunakan untuk analisis tekstur/ekstraksi ciri. GLCM merupakan suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra.

Koordinat pasangan piksel memiliki jarak d dan orientasi sudut θ . Jarak direpresentasikan dalam piksel dan sudut direpresentasikan dalam derajat. Orientasi

sudut terbentuk berdasarkan empat arah sudut yaitu, 0° , 45° , 90° dan 135° , dan jarak antar piksel sebesar 1 piksel.

Tahapan yang dilakukan pada perhitungan GLCM adalah sebagai berikut [4]:

1. Pembentukan matriks awal GLCM dari pasangan dua piksel yang berjajar sesuai dengan arah 0° , 45° , 90° atau 135° .
2. Membentuk matriks yang simetris dengan menjumlahkan matriks awal GLCM dengan nilai transposnya.
3. Menormalisasi matriks GLCM dengan membagi setiap elemen matriks dengan jumlah pasangan piksel.
4. Ekstraksi ciri,

Ekstraksi ciri adalah proses mengambil ciri-ciri yang terdapat pada suatu objek dalam citra. Pada proses ini objek di dalam citra dihitung properti-properti objek yang berkaitan sebagai ciri. Ciri-ciri tersebut adalah sebagai berikut :

1) *Contrast*

Merupakan representasi nilai terhadap variasi tingkat keabuan dalam matriks cooccurrence. Apabila suatu piksel dengan piksel tetangganya mempunyai nilai intensitas yang berdekatan, maka kontras tekstur sangat rendah.

Contrast :

$$\sum_{i_1} \sum_{i_2} (i_1 - i_2)^2 p(i_1, i_2)$$

2) *Homogeneity*

Berfungsi mengukur tingkat homogenitas atau kesamaan variasi dalam intensitas keabuan citra pada matriks co-occurrence. Homogenitas bernilai tinggi jika pasangan piksel mempunyai nilai keabuan yang seragam.

Homogeneity :

$$\sum_{i_1} \sum_{i_2} \frac{p(i_1, i_2)}{1 + |i_1 - i_2|}$$

3) *Energy*

Merupakan karakteristik untuk melihat tingkat keseragaman tekstur. Jika semakin tinggi nilai energi, maka tingkat homogenitas tekstur tinggi dan variasi intensitas dalam citra mengecil.

Energy :

$$\sum_{i_1} \sum_{i_2} p^2(i_1, i_2)$$

4) *Entropy*

Entropy merupakan kebalikan dari homogeneity yaitu ukuran tingkat keteracakan derajat keabuan suatu citra pada matriks cooccurrence. Nilai entropi memberikan informasi berupa fitur permukaan tekstur kasar ataupun halus. Apabila nilai entropi semakin mendekati 1 maka derajat kekasaran tekstur semakin tinggi begitupula sebaliknya.

Entropy :

$$\sum_{i_1} \sum_{i_2} p(i_1, i_2) \log p(i_1, i_2)$$

2. 2. 5. Metode *K-Nearest Neighbor* (K-NN)

Algoritma *K-Nearest Neighbor* (KNN) merupakan metode yang menggunakan algoritma *supervised* dimana hasil *query instance* diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Penggunaan *K-Nearest Neighbor* bertujuan untuk mengklasifikasikan objek baru berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek baru tersebut. Dalam hal ini jumlah data atau biasa disebut dengan tetangga terdekat ditentukan oleh user yang dinyatakan oleh k. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan *Euclidean Distance* [7][8].

Berikut adalah Langkah-langkah dalam menggunakan metode *K-Nearest Neighbor* [9] :

1. Menentukan parameter K (jumlah tetangga paling dekat)
2. Menghitung kuadrat jarak euclid (query instance) masing-masing obyek terhadap data sampel yang diberikan.

Rumus *euclidean distance* [10] :

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2}$$

Keterangan:

x_1 = Sampel Data

x_2 = Data Uji / Testing

i = Variabel Data

d = Jarak

p = Dimensi Data

3. Kemudian mengurutkan objek-objek tersebut kedalam kelompok yang mempunyai jarak euclid terkecil.
4. Mengumpulkan kategori Y (Klasifikasi *nearest neighbor*)
5. Dengan menggunakan kategori *nearest neighbor* yang paling mayoritas maka diprediksi nilai query instan yang telah dihitung.

Berikut adalah contoh penerapan perhitungan manual dari K-NN [11]:

Contoh data, terdapat beberapa data yang berasal dari survey questioner tentang klasifikasi jenis rempah-rempah berdasarkan fitur warna RGB tekstur. Proses pengujian klasifikasi dilakukan dengan menggunakan Algoritma *K-Nearest Neighbor* dengan nilai $K=1$, $K=3$ dan $K=5$, metode pengukuran jarak yang digunakan adalah Euclidean Distance. secara keseluruhan data training yang digunakan adalah 30 untuk masing-masing jenis rempah-rempah sehingga total keseluruhan berjumlah 120 data.

Tabel 2. 2 Sample Data Training

Nama	Red	Green	Blue	Energy	Homogeneity
Jahe 1	243.947	238.967	231.500	0,75108	0,98817
Jahe 2	250.102	247.829	244.378	0,84991	0,98974
Kencur 1	251.516	249.977	248.058	0,93425	0,99570
Kencur 2	247.648	245.583	243.968	0,90663	0,99389
Kunyit 1	216.296	207.255	200.842	0,56372	0,98040
Kunyit 2	198.486	184.850	176.708	0,34896	0,95906
Lengkuas 1	230.962	221.110	213.744	0,66720	0,99041
Lengkuas 2	248.632	244.912	240.438	0,86982	0,99466

Tabel 2. 3 Sample Data Testing

Nama	Red	Green	Blue	Energy	Homogeneity
Jahe 1	240.915	235.502	228.249	0,74449	0,98763
Jahe 6	249.537	246.454	240.995	0,80756	0,98988
Jahe 8	250.110	247.861	244.917	0,89477	0,99255
Kencur 1	248.707	246.5	244.557	0.90695	0.9934
Kencur 5	251.905	251.059	250.285	0,96216	0,99771
Kencur 6	250.997	249.764	248.898	0,94807	0,99628
Kunyit 3	244.339	241.190	237.806	0,83032	0,99002
Kunyit 5	247.306	244.755	242.526	0,86992	0,99196
Kunyit 8	247.402	244.262	241.149	0.8629	0.99367

Lengkuas 1	229.378	219.924	211.976	0,64473	0,98368
Lengkuas 2	237.728	230.507	223.552	0,73537	0,98993
Lengkuas 3	247.495	244.549	241.371	0.87983	0.99575

Setelah diperoleh data yang digunakan untuk pengujian kemudian dilakukan proses klasifikasi menggunakan Algoritma *K-Nearest Neighbour* dengan langkah-langkah sebagai berikut :

1. Menentukan parameter k (jumlah tetangga paling dekat). Parameter k (jumlah tetangga paling dekat untuk melakukan klasifikasi jenis rempahrempah diatas adalah menggunakan nilai k=1, k=3, k=5.
2. Menghitung kuadrat jarak euclidean masing-masing objek terhadap data sample yang diberikan.

Sebagai contoh perhitungan diambil data testing yang pertama yaitu data testing Jahe1, lalu dikalikan terhadap seluruh data training yang ada:

$$d = \sqrt{(243.947 - 240.915)^2 + (238.967 - 235.502)^2 + (231.500 - 228.249)^2 + (0,75108 - 0,74449)^2 + (0,98817 - 0,98763)^2} = 5636,33303$$

Jadi jarak Euclidean Distance yang diperoleh adalah sebesar 5636,33303. Data testing jahe1 tersebut dihitung Euclidean Distance nya keseluruhan data training yang ada, kemudian dirangking dengan mengambil 5 neighbor yang terdekat, hasil dari perkalian data testing jahe1 terhadap 120 data training dapat dilihat pada tabel berikut ini.

Tabel 2. 4 Sampel Hasil Pengujian

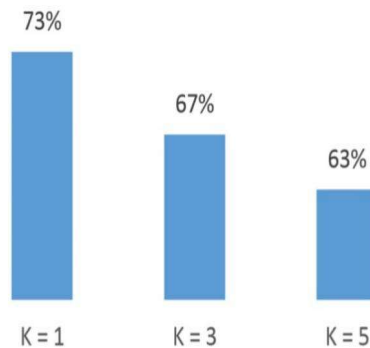
Ranking	Euclidean Distance	Nama Kelas
1	1784,19422	Jahe
2	1872,70873	Lengkuas
3	1938,29461	Lengkuas
4	2042,767731	Lengkuas
5	2174,98	Jahe

Dari hasil pengujian diatas dapat dilihat bahwa jika menggunakan nilai $K=1$ maka proses klasifikasi benar, jika menggunakan nilai $K=3$ maka proses klasifikasi salah, dan jika menggunakan nilai $K=5$ maka proses klasifikasi juga salah. Pada penelitian ini dilakukan 30 kali pengujian dengan menggunakan data training masing-masing jenis rempah-rempah terhadap 120 data training, hasil dari keseluruhan pengujian dapat dilihat pada table 2.5.

Tabel 2. 5 Hasil Pengujian

Nilai K	Jumlah Kebenaran	Persentasi Kebenaran
1	23/30	73%
3	20/30	67%
5	19/30	63%

Untuk lebih jelasnya perbandingan hasil pengujian dapat dilihat pada diagram seperti yang ditunjukkan pada gambar 2.2 :



Gambar 2. 2 Diagram Hasil Pengujian

Dari hasil pengujian diatas dapat dilihat bahwa nilai K terbaik untuk melakukan klasifikasi jenis rempah-rempah berdasarkan fitur warna RGB dan tekstur adalah 1 dengan nilai akurasi 73%, yang kedua adalah nilai $K=3$ dengan nilai akurasi 67% dan yang ketiga adalah nilai $K=5$ dengan nilai akurasi 63%.

2. 2. 6. Ekstrasi Fitur

Ekstrasi fitur merupakan suatu pengambilan ciri atau fitur dari suatu bentuk nantinya nilai yang didapatkan akan dianalisis untuk proses selanjutnya. Fitur ialah karakteristik untuk dari suatu objek yang berguna untuk proses klasifikasi. cara melakukan ekstrasi fitur ialah dengan menghitung jumlah titik atau *pixel* yang ada pada citra [12].

2. 2. 7. Perangkat Lunak Pendukung

Tabel 2. 6 Perangkat Lunak Pendukung

No	Perangkat Lunak Pendukung	Kegunaan
1.	Mendeley Dekstop	Untuk membuat daftar Pustaka secara otomatis.
2.	Phyton	Bahasa Pemrograman untuk membuat sebuah program

2. 2. 8. Evaluasi Model

Evaluasi model dalam penelitian ini dilakukan untuk menguji performansi dari sistem yang dibangun. Beberapa cara untuk melakukan evaluasi adalah dengan menghitung akurasi (*accuracy*), presisi (*precision*), dan *recall* dari hasil pengolahan. Metode yang digunakan untuk perhitungan akurasi adalah *confusion matrix* untuk mengevaluasi kinerja model atau algoritma.

Akurasi dalam klasifikasi adalah persentase ketepatan record data yang diklasifikasikan secara benar setelah dilakukan pengujian pada hasil klasifikasi. Presisi atau *confidence* adalah proporsi kasus yang diprediksi positif yang juga positif benar pada data yang sebenarnya. *Recall* atau *sensitivity* adalah proporsi kasus positif yang sebenarnya yang diprediksi positif secara benar [13].

2. 2. 9. *Confusion Matrix*

Confusion matrix adalah suatu metode yang digunakan untuk melakukan perhitungan akurasi pada konsep data mining. Evaluasi dengan *confusion matrix* menghasilkan nilai akurasi, presisi dan *recall*. *Confusion matrix* dapat disajikan seperti pada Tabel berikut [14]:

Tabel 2. 7 *Confusion Matrix*

Kelas	Terklasifikasi Positif	Terklasifikasi Negatif
Positif	TP (<i>True Positive</i>)	FN (<i>False Negative</i>)
Negatif	FP (<i>False Positive</i>)	TN (<i>True Negative</i>)

Keterangan :

1. *True Negative* (TN) merupakan jumlah data negatif yang terdeteksi dengan benar oleh sistem,
2. *False Positive* (FP) merupakan data negatif namun terdeteksi sebagai data positif oleh sistem,
3. *True Positive* (TP) merupakan data positif yang terdeteksi benar oleh sistem.
4. *False Negative* (FN) merupakan data negative yang terdeteksi salah oleh sistem.

Berikut persamaan model konfussion matrix :

1. Nilai Akurasi (*accuracy*) menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. Persamaan Akurasi sebagai berikut.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\%$$

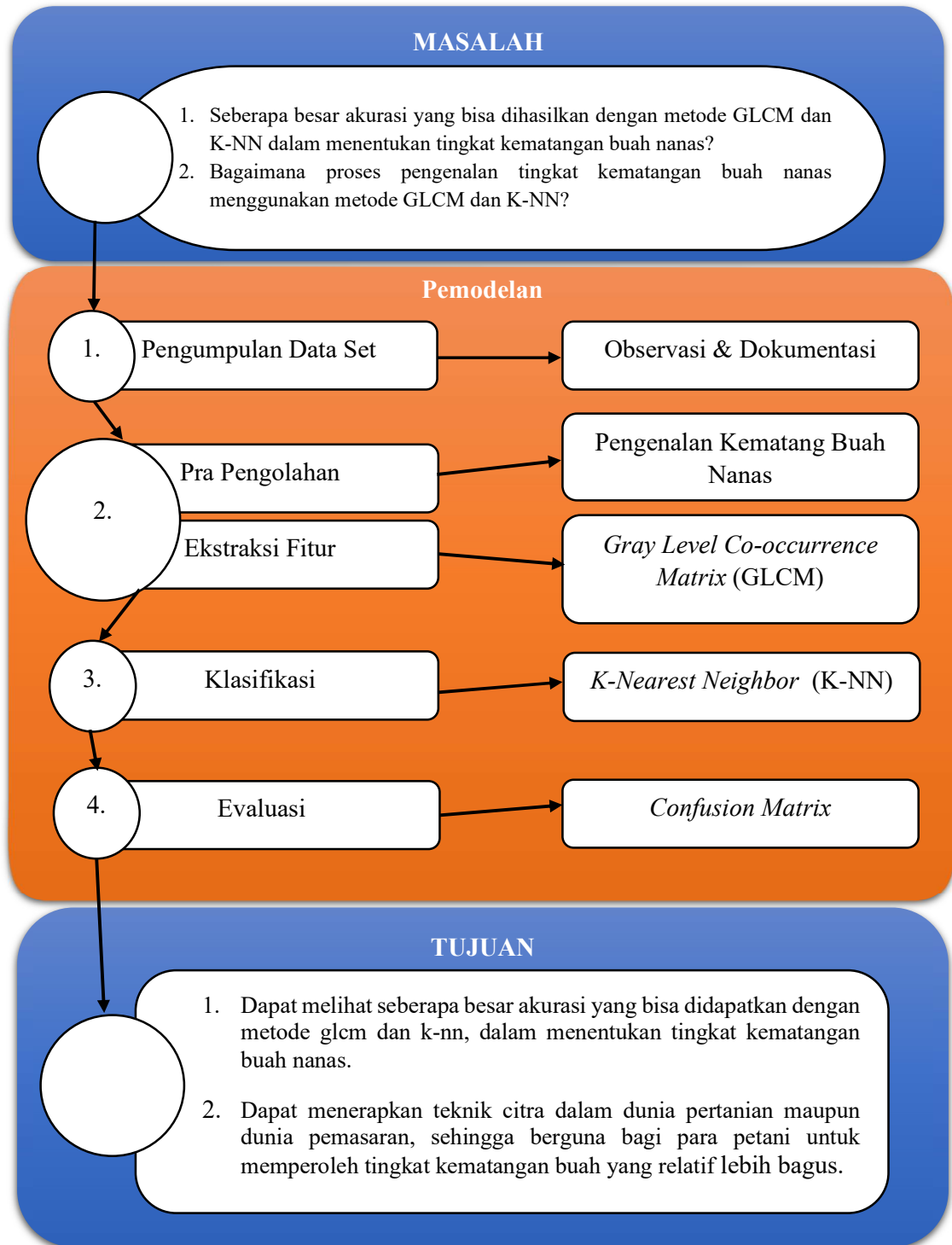
2. Nilai Presisi (*precision*) menggambarkan jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif. Persamaan Presisi sebagai berikut.

$$Precision = \frac{TP}{FP + TP} * 100\%$$

3. Nilai Sensitivity (*recall*) menunjukkan berapa persen data kategori positif yang terklasifikasikan dengan benar oleh system. Persamaan Recall sebagai berikut.

$$Recall = \frac{TP}{FN + TP} * 100\%$$

2. 3. Kerangka Pikir



Gambar 2. 3 Kerangka Pikir

BAB III

METODE PENELITIAN

3. 1. Jenis, Metode, Subjek, Objek, Waktu dan Lokasi Penelitian

Dipandang dari tingkat penerapannya, maka penelitian ini merupakan penelitian terapan. Dipandang dari jenis informasi yang diolah, maka penelitian ini merupakan penelitian kuantitatif. Dipandang dari perlakuan terhadap data maka penelitian ini merupakan penelitian konfirmatori.

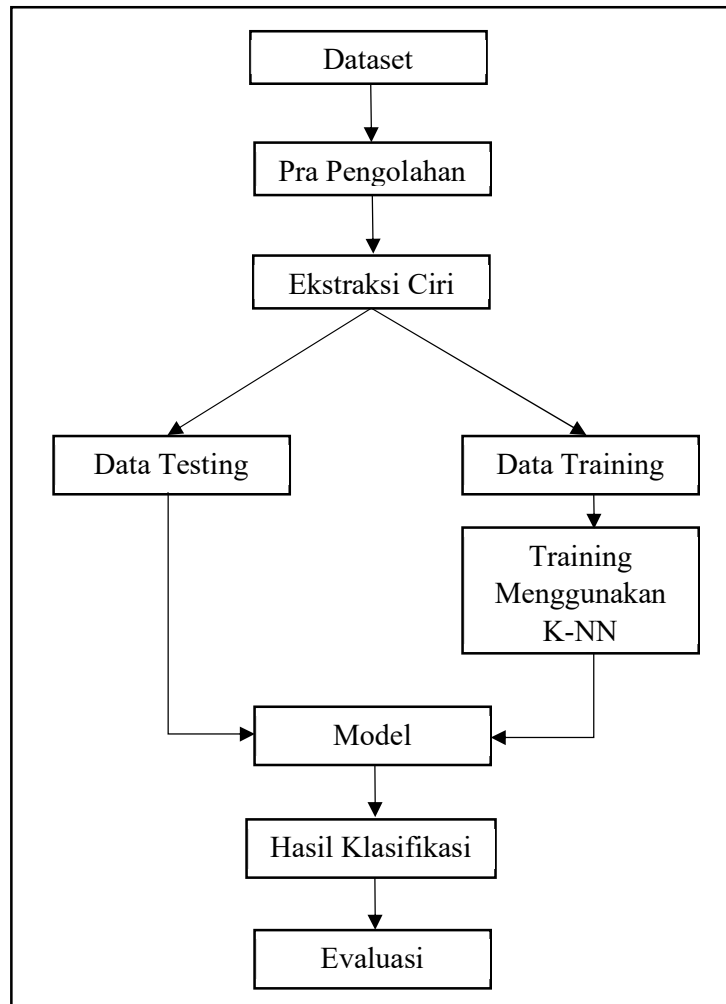
Penelitian ini menggunakan metode penelitian eksperimen. Dengan demikian jenis penelitian ini adalah penelitian eksperimental.

Subjek penelitian ini pengenalan tingkat kematangan pada objek buah nanas. Penelitian ini dimulai dari September 2020 sampai dengan November 2020 yang berlokasi di Desa Lobong, Kec. Passi Barat, Kab. Bolaang Mongondow.

3. 2. Pengumpulan Data

Data primer penelitian ini adalah citra Buah Nanas, dimana peneliti mengambil gambar secara langsung menggunakan Kamera Smartphone Vivo Z1 Pro dengan resolusi 16MP. Jumlah data yang digunakan yaitu 209 file dalam format JPG dengan tingkat kematangan (Mentah dan Matang).

3.3. Pemodelan



Gambar 3.1 Pemodelan

3.3.1. Pra Pengolahan Data

Sebelum mengolah data, pertama peneliti melakukan proses *histogram equalization*. Ini dilakukan karena *histogram equalization* merupakan proses yang akan mengubah nilai distribusi nilai derajat keabuan pada suatu citra sehingga akan menjadi seragam.

3.3.2. Ekstraksi Ciri

Ekstraksi ciri memiliki fungsi sebagai pendeteksi ciri dari sebuah citra. Ciri yang bisa digunakan untuk membedakan citra satu dan citra lain, diantaranya ialah ciri bentuk, ukuran, geometri, tekstur, dan warna. Pada penelitian kali ini menggunakan *Gray Level Co-Occurrence Matrix*. Masing-masing citra diekstrak cirinya berdasarkan parameter-parameter tertentu dan dikelompokkan pada kelas tertentu. Kemudian nilai dari parameter tersebut dijadikan sebagai data *input* dalam proses identifikasi klasifikasi.

3.3.3. Data Training

Data training berupa data yang telah terekstrak cirinya yang selanjutnya akan dilatih menggunakan algoritma *K-Nearest Neighbor*, jumlah data training yang digunakan berupa 10% dari sample citra buah nanas dengan tingkat kematangan yang beda. Data training ini berupa hasil dari ekstraksi *Gray Level Co-occurrence Matrix* terhadap pengenalan citra nanas.

3.3.4. Training Menggunakan K-NN

Training menggunakan K-NN yakni menjadikan data training sebagai data inputan. Dimana algoritma K-NN bekerja berdasar pada jarak terpendek dari training dan test dalam penentuannya, kemudian mengambil mayoritas hasil ketentuan yang telah didapatkan untuk dijadikan prediksi dari test.

3.3.5. Model

Model adalah hasil dari proses training dari algoritman *K-Nearest Neighbor* menggunakan data training.

3.3.6. Data Testing

Data testing merupakan data yang telah terekstrasi cirinya, dan digunakan untuk menguji data yang telah dilatih. Jumlah data yang digunakan berupa 90% dari sampel citra buah nanas dengan tingkat kematangan yang berbeda.

3.3.7. Hasil Klasifikasi

Hasil klasifikasi merupakan nilai *output*, yang didapat pada data testing dari proses klasifikasi menggunakan algoritma K-NN berdasarkan model yang diperoleh dari data traning.

3.3.8. Evaluasi

Evaluasi bertujuan untuk mengetahui hasil kinerja dari metode yang digunakan. Model yang telah dihasilkan kemudian dievaluasi menggunakan *confusion matrix* untuk mengetahui tingkat akurasinya.

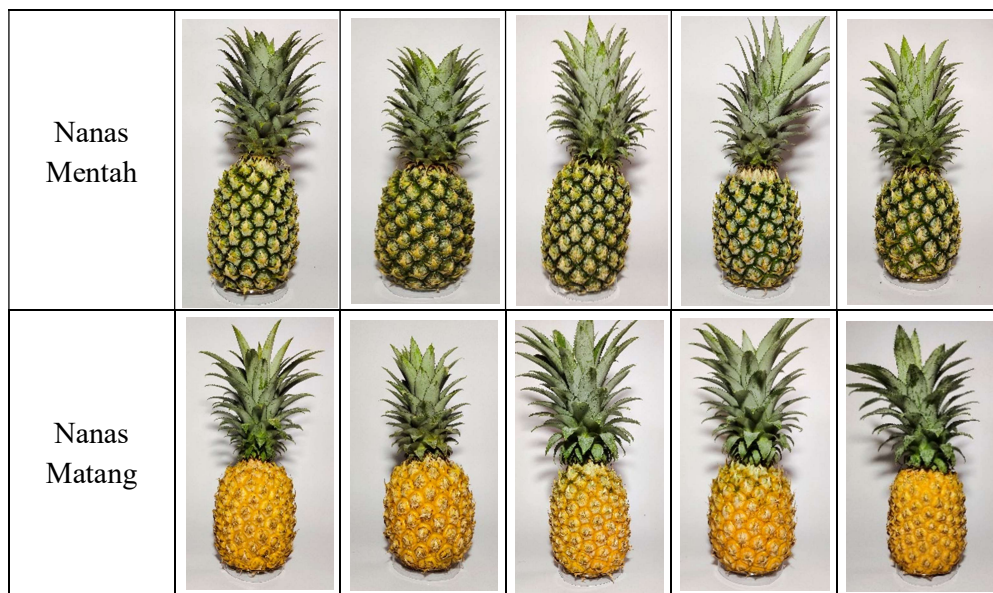
BAB IV

HASIL PENELITIAN

4.1 Hasil Pengumpulan Data

Dataset yang digunakan dalam penelitian ini adalah data real dengan cara mengambil foto secara langsung menggunakan kamera Vivo Z1 Pro dengan resolusi kamera 16 Megapixel. Dengan jumlah data 239 citra buah Nanas yang terdiri dari 30 data testing dan 209 data training dengan tingkat kematangan yang berbeda (115 Matang, 94 Mentah).

Tabel 4. 1 Data Gambar Buah Nanas



4.2 Hasil Pemodelan

4.2.1 Pengambilan Citra Buah Nanas

Pada penelitian ini, dataset yang digunakan adalah citra buah nanas. Data yang akan digunakan adalah sebanyak 239 data dengan rincian 30 data testing, 115 citra untuk buah nanas yang sudah matang dan 94 citra untuk buah yang masih mentah. Pengambilan citra dilakukan dengan menggunakan alas kertas HVS yang memiliki warna latar putih (*background*). Warna latar putih bertujuan untuk

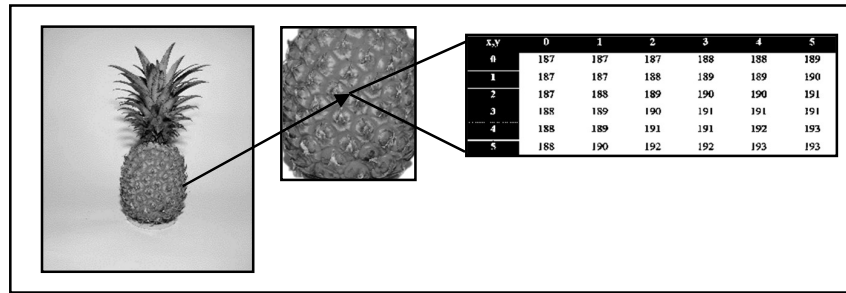
mempermudah proses pra pengolahan dan ekstraksi ciri dimana warna latar tidak mempengaruhi nilai dari masing masing nilai ciri citra.

4.2.2 Pra-Pengolahan

Pra-pengolahan citra (*image pre-processing*), yakni proses paling awal dalam pengolahan citra sebelum proses utama dilakukan. Pada tahap ini citra buah nanas yang ada dikonversi agar mendapatkan data citra nanas yang sesuai kebutuhan. Tahap ini berfungsi untuk menormalisasi citra dari permasalahan luminasi yang teralalu gelap atau terlalu terang sehingga dapat meningkatkan performansi dari sistem deteksi tingkat kematangan.

Proses awal yang dilakukan yaitu merubah citra *training* atau citra *testing* yang awalnya citra dari RGB menjadi citra *grayscale*, perubahan ini dilakukan karena citra *grayscale* memiliki persamaan yang sederhana dan mampu mengurangi kebutuhan memory dimana nilai warna putih diwakili dengan angka 255 dan nilai warna hitam diwakili dengan angka 0. Setelah citra asli dikonversi ke citra *grayscale* maka pra-pengolahan selanjutnya adalah normalisasi citra dengan histogram ekualisasi. Histogram ekualisasi adalah proses yang mengonversikan bagian nilai derajat keabuan pada sebuah citra sehingga menjadi sejenis. Maksud dari proses *histogram equalization* adalah untuk mencapai diseminasi histogram yang menyeluruh sehingga setiap derajat keabuan mendapat jumlah pixel yang relatif sama.

Proses selanjutnya yakni segmentasi citra, bertujuan untuk membagi suatu citra menjadi wilayah-wilayah yang homogen berdasarkan kriteria keserupaan yang tertentu antara tingkat keabuan suatu piksel dengan tingkat keabuan piksel-piksel tetangganya. Tahapan ini bertujuan untuk mempartisi citra menjadi bagian-bagian pokok yang mengandung informasi penting, misalnya memisahkan objek dan latar belakang. Hasil dari proses segmentasi citra akan digunakan untuk proses tingkat tinggi lebih lanjut yang dapat dilakukan terhadap suatu citra, misalnya proses klasifikasi citra dan proses identifikasi objek.



Gambar 4. 1 Contoh ilustrasi pengambilan *pixel* pada citra *Grayscale*

Tabel 4. 2 Citra Grayscale 6x6

x,y	0	1	2	3	4	5
0	187	187	187	188	188	189
1	187	187	188	189	189	190
2	187	188	189	190	190	191
3	188	189	190	191	191	191
4	188	189	191	191	192	193
5	188	190	192	192	193	193

Setelah didapat hasil diatas, maka dilakukan perhitungan dengan langkah dan rumus pada tabel dibawah :

Tabel 4. 3 Perhitungan dengan Langkah dan rumus

Derajat Grayscale (j)	Kemunculan	$s = \sum_{j=0}^k \frac{n_j}{n}$
187	6	6/36
188	7	7/36
189	6	6/36
190	5	5/36
191	6	6/36
192	3	3/36
193	3	3/36

Hasil dari perhitungan Perataan Histogram diatas dapat disimpulkan seperti berikut:

Tabel 4. 4 Citra asli 6x6

187	187	187	188	188	189
187	187	188	189	189	190
187	188	189	190	190	191
188	189	190	191	191	191
188	189	191	191	192	193
188	190	192	192	193	193

Tabel 4. 5 Data citra 6x6 setelah dilakukan Histogram Equalization

0,1667	0,1667	0,1667	0,1944	0,1944	0,1667
0,1667	0,1667	0,1944	0,1667	0,1667	0,1389
0,1667	0,1944	0,1667	0,1389	0,1389	0,1667
0,1944	0,1667	0,1389	0,1667	0,1667	0,1667
0,1944	0,1667	0,1667	0,1667	0,0833	0,0833
0,1944	0,1389	0,0833	0,0833	0,0833	0,0833

4.2.3 Ekstraksi Fitur

Fitur ekstrasi *Gray Level Co-onccurrence Matrix* (GLCM) merupakan *matrix* yang menggambarkan frekuensi munculnya pasangan dua pixel dengan intensitas tertentu dalam jarak dapat berupa nilai 1,2,3 dan seterusnya sedangkan orientasi arah sudut dapat berupa 0° , 45° , 90° , 135° .

Contoh : Nilai pada jarak 1 dan orientasi arah sudut 0°

Nilai pada table 4.6 berupa matrix 4 x 4, dari matrix 255 x 255 yang disederhanakan untuk memudahkan perhitungan.

Tabel 4. 6 Nilai dari suatu citra

181	181	181	182
181	181	182	183
181	182	183	184
183	183	184	186

Setelah menentukan arahnya selanjutnya membentuk matriks kookurens dengan cara menghitung frekuensi kemunculan pasangan nilai keabuan piksel referensi dan piksel tetangga pada jarak dan arah yang ditentukan. Perhitungan menggunakan nilai orientasi dengan arah sudut 0^0 . 0^0 yaitu nilai gambar yang memiliki nilai yang sama dan bersebelahan. Hasilnya sebagai berikut:

Tabel 4. 7 Matrix GLCM

i/j	181	182	183	184	186
181	3	3	0	0	0
182	3	0	2	0	0
183	0	2	1	2	0
184	0	0	2	0	1
186	0	0	0	1	0

Jumlah Total Nilai GLCM = 20

Langkah selanjutnya yaitu menormalisasikan matrix dengan cara menjumlahkan seluruh nilai yang ada dalam matriks tersebut.

Tabel 4. 8 Matrix Normalisasi

i/j	181	182	183	184	186
181	0,15	0,15	0	0	0
182	0,15	0	0,1	0	0
183	0	0,1	0,05	0,1	0
184	0	0	0,1	0	0,05
186	0	0	0	0,05	0

Langkah terakhir adalah menghitung ciri statistic GLCM:

Terdapat total 28 fitur GLCM, tetapi hanya 14 fitur yang dapat dikalkulasikan dengan metode GLCM diantaranya adalah *Angular second moment, Correlation, Contrast, Variance, Sum Variance, Inverse Difference Moment, Sum Entropy, Entropy, Max, Correlation Coeff, Difference Entropy, Difference Variance, Sum Average Info Measure of Correlation 1 dan 2*. Dan dari 14 fitur tersebut inilah fitur utamanya yakni *Contrast Energy, entropy, dan homogeneity*.

a. Nilai *Contras*

Contras =

$$(181-181)^2(0,15) + (181-182)^2(0,15) + (182-181)^2(0,15) + (182-183)^2(0,1) + \\ (183-182)^2(0,1) + (183-183)^2(0,05) + (183-184)^2(0,1) + (184-183)^2(0,1) + \\ (184-186)^2(0,05) + (186-184)^2(0,05)$$

Contras =

$$0 + 0,15 + 0,15 + 0,1 + 0,1 + 0 + 0,1 + 0,1 + 0,2 + 0,2$$

$$\text{Contras} = 1,1$$

b. Nilai *Energy*

Energy =

$$(0,15)^2 + (0,15)^2 + (0,15)^2 + (0,1)^2 + (0,1)^2 + (0,05)^2 + (0,1)^2 + (0,1)^2 + (0,05)^2 + \\ (0,05)^2$$

Energy =

$$(0,0225) + (0,0225) + (0,0225) + (0,01) + (0,01) + (0,0025) + (0,01) + (0,01) + \\ (0,0025) + (0,0025)$$

$$\text{Energy} = 0,115$$

c. Nilai *Homogeneity*

Homogeneity =

$$((0,15)/1 + (181-181)) + ((0,15)/1 + (181-182)) + ((0,15)/1 + (182-181)) + \\ ((0,1)/1 + (182-183)) + (0,1)/1 + (183-182)) + (0,05)/1 + (183-183)) + \\ ((0,1)/1 + (183-184)) + ((0,1)/1 + (184-183)) + ((0,05)/1 + (184-186)) +$$

$$((0,05)/1 + (186-184))$$

$$Homogeneity =$$

$$(0,15 + 0) + (0,15 + -1) + (0,15 + 1) + (0,1 + -1) + (0,1 + 1) + (0,05 + 0) +$$

$$(0,1 + -1) + (0,1 + 1) + (0,05 + -2) + (0,05 + 2)$$

$$Homogeneity =$$

$$(0,15) + (0,85) + (1,15) + (-0,9) + (1,1) + (0,05) + (-0,9) + (1,1) + (-1,95) + (2,05)$$

$$Homogeneity = 2,7$$

d. Nilai *Entropy*

$$Entropy =$$

$$(0,15 \log 0,15) + (0,15 \log 0,15) + (0,15 \log 0,15) + (0,1 \log 0,1) +$$

$$(0,1 \log 0,1) + (0,05 \log 0,05) + (0,1 \log 0,1) + (0,1 \log 0,1) + (0,05 \log 0,05) +$$

$$(0,05 \log 0,05)$$

$$Entropy =$$

$$(0,1235) + (0,1235) + (0,1235) + (-0,1) + (-0,1) + (-0,0650) + (-0,1) + (-0,1) + (-$$

$$0,0650)$$

$$Entropy = -0.2245$$

Empat ciri yang di dapatkan dari matrix CO-Occurrence inilah yang akan digunakan untuk tahap klasifikasi.

4.2.4 Klasifikasi

Pada tahapan ini hasil dari ekstraksi fitur data testing dan data uji pada pembahasan sebelumnya diklasifikasikan berdasarkan kelasnya masing-masing dengan jumlah $k=1$, $k=3$, $k=5$, $k=7$, dan $k=9$, nilai dari jumlah k tersebut diambil berdasarkan nilai angka ganjil agar dapat menghindari hasil klasifikasi dengan jumlah data yang seimbang. Nilai k tersebut juga dapat menentukan bagaimana kinerja algoritma K-NN, dalam melakukan pengenalan kematangan buah nanas, seperti yang terlihat pada tabel dibawah ini:

Tabel 4. 9 Hasil Ekstraksi Fitur

Data Trainning	X1	X2	X3	Kelas
0	345.46010965119564	0.48409003545776763	0.672895255149799	Matang
1	369.15543624161114	0.49879092185650237	0.6563064090328433	Matang
2	414.2513199105151	0.4996322887533262	0.6548244725614476	Matang
3	483.1694407158842	0.4976289742762354	0.6294061963646077	Matang
4	0.6294061963646077	0.4970691352594656	0.6887738955870795	Matang
...	...	...	...	...
115	446.3613383628081	0.45821151627299045	0.6829291990302365	Mentah
116	432.7046000097978	0.45999692868662984	0.6837467768795326	Mentah
117	579.1459874949803	0.5270229999841366	0.6771599973419347	Mentah
118	468.7663626455571	0.5245632287236388	0.6857888272832654	Mentah
119	392.43284116331137	0.5116809108942975	0.7172305467730032	Mentah
120	396.4266219239377	0.49852836795878086	0.7128763020548378	Mentah
...	...	...	...	...
Data Testing	428.0242505592846	0.5006621613251726	0.6837593512750564	?

Tabel diatas adalah hasil dari 209 data training dan ekstraksi fitur dari salah satu citra dari 30 data testing yang tidak diketahui kelasnya. Setelah itu setiap jarak dari data training ke data testing dihitung menggunakan rumus perhitungan jarak euclidean distance seperti dibawah ini:

$$\begin{aligned}
 d &= \sqrt{(345,4601 - 428,0242)^2 + (0,4840 - 0,5006)^2 + (0,6728 - 0,6837)^2} \\
 &= \sqrt{(-82,5641)^2 + (-0,0166)^2 + (-0,0109)^2} \\
 &= \sqrt{6,816,83060881 + 0,00027556 + 0,00011881} \\
 &= \sqrt{6,8172249788} = 2,6109816121
 \end{aligned}$$

Kemudian dari proses perhitungan jarak yang telah dilakukan maka nilai yang didapatkan dari perhitungan jarak antara data training dengan data testing tersebut akan disusun berdasarkan nilai pada masing-masing kelas seperti pada tabel 4.10 sebagai berikut:

Tabel 4. 10 Perhitungan Jarak *Euclidean Distance*

Data Training	Perhitungan jarak <i>Euclidean Distance</i>	Nilai
0	$d = \sqrt{(345,4601 - 428,0242)^2 + (0,4840 - 0,5006)^2 + (0,6728 - 0,6837)^2}$	2,6109816121
1	$d = \sqrt{(369,1554 - 428,0242)^2 + (0,4987 - 0,5006)^2 + (0,6563 - 0,6837)^2}$	1,8617975141
2	$d = \sqrt{(414,2513 - 428,0242)^2 + (0,4996 - 0,5006)^2 + (0,6548 - 0,6837)^2}$	13,772930357
3	$d = \sqrt{(483,1694 - 428,0242)^2 + (0,4976 - 0,5006)^2 + (0,6294 - 0,6837)^2}$	1,7446921141
4	$d = \sqrt{(0,6294 - 428,0242)^2 + (0,4970 - 0,5006)^2 + (0,6887 - 0,6837)^2}$	13,5154116854
...	...	...
115	$d = \sqrt{(446,3613 - 428,0242)^2 + (0,5821 - 0,5006)^2 + (0,6829 - 0,6837)^2}$	18,3424310791
116	$d = \sqrt{(432,7046 - 428,0242)^2 + (0,4599 - 0,5006)^2 + (0,6837 - 0,6837)^2}$	4,6805737796
117	$d = \sqrt{(579,1459 - 428,0242)^2 + (0,5270 - 0,5006)^2 + (0,6771 - 0,6837)^2}$	4,778952364
118	$d = \sqrt{(468,7663 - 428,0242)^2 + (0,5245 - 0,5006)^2 + (0,6857 - 0,6837)^2}$	1,2886015375
119	$d = \sqrt{(392,4328 - 428,0242)^2 + (0,5116 - 0,5006)^2 + (0,7172 - 0,6837)^2}$	1,1260510663
120	$d = \sqrt{(396,4266 - 428,0242)^2 + (0,4985 - 0,5006)^2 + (0,7128 - 0,6837)^2}$	31,5976134697
...	...	...

Setelah nilai jarak dari masing-masing data latih ke data uji didapatkan maka akan disusun berdasarkan nilai terkecil berikut.

Tabel 4. 11 Hasil Pengurutan Data

Data	Jarak ke masing-masing data latih	Kelas
119	1,12605	Mentah
118	1,28860	Mentah
3	1,74469	Matang
1	1,86179	Matang
59	1,91881	Matang
151	2,01233	Mentah
0	2,61098	Matang
15	2,72709	Matang
67	3,06305	Matang
125	4,38672	Mentah
...	...	...

Tabel diatas adalah hasil pengurutan data dari nilai terkecil berdasarkan perhitungan jarak dari masing-masing data latih ke data uji. Dari hasil pengurutan yang terdapat pada tabel 4.11, maka yang akan diambil adalah nilai berdasarkan jumlah k yang telah ditentukan sebelumnya, yakni $k=1$, $k=3$, $k=5$, $k=7$, dan $k=9$ seperti berikut:

Tabel 4. 12 Nilai $K=1$

Data Ke	Jarak ke masing-masing data latih	Kelas
119	1,12605	Mentah

Dari data uji $k=1$, dapat dilihat bahwa kelas yang muncul adalah “Mentah” dengan 1 kelas mentah seperti tabel dibawah ini :

Tabel 4. 13 Kelas data uji dari nilai $K=1$

Kelas	Jumlah Data
Matang	0
Mentah	1

Tabel 4. 14 Nilai $K=3$

Data Ke	Jarak ke masing-masing data latih	Kelas
119	1,12605	Mentah
118	1,28860	Mentah
3	1,74469	Matang

Dari data uji nilai $k=3$, dapat dilihat bahwa kelas yang paling banyak muncul adalah “Mentah”, dengan 2 kelas mentah dan 1 kelas matang seperti tabel dibawah ini :

Tabel 4. 15 Kelas data uji dari nilai $K=3$

Kelas	Jumlah Data
Matang	1
Mentah	2

Tabel 4. 16 Nilai K=5

Data Ke	Jarak ke masing-masing data latih	Kelas
119	1,12605	Mentah
118	1,28860	Mentah
3	1,74469	Matang
1	1,86179	Matang
59	1,91881	Matang

Dari data uji nilai $k=5$, dapat dilihat bahwa kelas yang paling banyak muncul adalah “Matang “, dengan 3 kelas matang dan 2 kelas mentah seperti tabel dibawah ini :

Tabel 4. 17 Kelas data uji dari nilai $k=5$

Kelas	Jumlah Data
Matang	3
Mentah	2

Tabel 4. 18 Nilai K=7

Data Ke	Jarak ke masing-masing data latih	Kelas
119	1,12605	Mentah
118	1,28860	Mentah
3	1,74469	Matang
1	1,86179	Matang
59	1,91881	Matang
151	2,01233	Mentah
0	2,61098	Matang

Dari data uji nilai $k=7$, dapat dilihat bahwa kelas yang paling banyak muncul adalah “Matang “, dengan 4 kelas matang dan 3 kelas mentah seperti tabel dibawah ini:

Tabel 4. 19 Kelas data uji dari nilai K=7

Kelas	Jumlah Data
Matang	4
Mentah	3

Tabel 4. 20 Nilai K=9

Data Ke	Jarak ke masing-masing data latih	Kelas
119	1,12605	Mentah
118	1,28860	Mentah
3	1,74469	Matang
1	1,86179	Matang
59	1,91881	Matang
151	2,01233	Mentah
0	2,61098	Matang
15	2,72709	Matang
67	3,06305	Matang

Dari data uji nilai k=9, dapat dilihat bahwa kelas yang paling banyak muncul adalah “Matang “, dengan 6 kelas matang dan 3 kelas mentah seperti tabel dibawah ini:

Tabel 4. 21 Kelas data uj darii nilai K=9

Kelas	Jumlah Data
Matang	6
Mentah	3

Dari penentuan jumlah k=1, k=3, k=5, k=7 dan k=9, maka dapat dinyatakan bahwa kelas yang paling mayoritas/mirip dengan data uji adalah “Matang”.

Tabel 4. 22 Hasil perhitungan *Euclidean distance* untuk data testing

Data Trainning	X1	X2	X3	Kelas
0	345.46010965119564	0.48409003545776763	0.672895255149799	Matang
1	369.15543624161114	0.49879092185650237	0.6563064090328433	Matang
2	414.2513199105151	0.4996322887533262	0.6548244725614476	Matang
3	483.1694407158842	0.4976289742762354	0.6294061963646077	Matang
4	0.6294061963646077	0.4970691352594656	0.6887738955870795	Matang
...	...	...	...	...
115	446.3613383628081	0.45821151627299045	0.6829291990302365	Mentah
116	432.7046000097978	0.45999692868662984	0.6837467768795326	Mentah
117	579.1459874949803	0.5270229999841366	0.6771599973419347	Mentah
118	468.7663626455571	0.5245632287236388	0.6857888272832654	Mentah
119	392.43284116331137	0.5116809108942975	0.7172305467730032	Mentah

120	396.4266219239377	0.49852836795878086	0.7128763020548378	Mentah
...	...	...	...	...
Data Testing	428.0242505592846	0.5006621613251726	0.6837593512750564	Matang

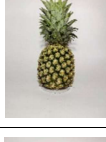
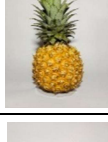
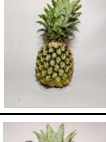
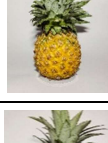
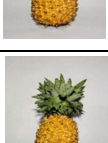
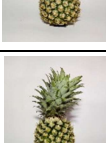
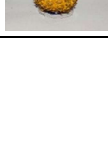
Hasil yang didapatkan dari perhitungan jarak Euclidean distance untuk data uji adalah “matang”

4.2.5 Evaluasi Model

Pada penelitian ini peneliti menggunakan confusion matriks sebagai metode untuk mengukur hasil akurasi dari metode klasifikasi yang telah diujikan sebelumnya. Berikut merupakan sebuah tabel dari hasil perhitungan 30 data uji dengan jumlah $k=1$, $k=3$, $k=5$, $k=7$ dan $k=9$ sebagai berikut:

Tabel 4. 23 Hasil data testing dari salah satu nilai K ($k=1$)

NO.	Gambar Uji	Klasifikasi	Hipotesis	Gambar	Klasifikasi	Hipotesis
1.		Matang	Matang		Mentah	Mentah
2.		Matang	Matang		Mentah	Matang
3.		Matang	Matang		Mentah	Matang
4.		Matang	Matang		Mentah	Mentah
5.		Matang	Matang		Mentah	Mentah

NO.	Gambar Uji	Klasifikasi	Hipotesis	Gambar	Klasifikasi	Hipotesis
6.		Matang	Matang		Mentah	Mentah
7.		Matang	Matang		Mentah	Mentah
8.		Matang	Matang		Mentah	Mentah
9.		Matang	Mentah		Mentah	Mentah
10.		Matang	Mentah		Mentah	Mentah
11.		Matang	Matang		Mentah	Mentah
12.		Matang	Matang		Mentah	Mentah
13.		Matang	Mentah		Mentah	Mentah
14.		Matang	Matang		Mentah	Mentah
15.		Matang	Mentah		Mentah	Mentah

Keterangan Warna :

 Data uji yang benar

 Data uji yang salah

Tabel 4. 24 Hasil data uji k=1

Kelas	Matang	Mentah
Matang	11	4
Mentah	2	13

Jumlah data yang diklasifikasikan benar (kotak kuning) = 24, jumlah data yang diklasifikasikan salah (kotak hijau) = 6.

Keterangan:

(TP) *True Positive* / Merupakan buah nanas matang yang terdeteksi matang oleh sistem = 11

(TN) *True Negative* / Merupakan buah nanas mentah yang terdeteksi mentah oleh sistem = 13

(FP) *False Positive* / Merupakan buah nanas mentah yang dideteksi matang oleh sistem = 2

(FN) *False Negative* / Merupakan buah nanas matang yang dideteksi mentah oleh sistem = 4

$$Akurasi = \frac{TP + TN}{TP + FP + FN + TN} \times 100\%$$

$$Akurasi = \frac{24}{30} \times 100\% = 80\%$$

Jadi, hasil data uji K-1 mendapatkan akurasi 80%.

Catatan : Nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data.

Tabel 4. 25 Hasil data uji k=3

Kelas	Matang	Mentah
Matang	12	3
Mentah	4	11

Jumlah data yang diklasifikasikan benar (kotak Kuning) = 23, jumlah data yang diklasifikasikan salah (kotak hijau) = 7.

Keterangan:

(TP) *True Positive* / Merupakan buah nanas matang yang terdeteksi matang oleh sistem = 12

(TN) *True Negative* / Merupakan buah nanas mentah yang terdeteksi mentah oleh sistem = 11

(FP) *False Positive* / Merupakan buah nanas mentah yang dideteksi matang oleh sistem = 4

(FN) *False Negative* / Merupakan buah nanas matang yang dideteksi mentah oleh sistem = 3

$$Akurasi = \frac{TP + TN}{TP + FP + FN + TN} \times 100\%$$

$$Akurasi = \frac{23}{30} \times 100\% = 76\%$$

Jadi, hasil data uji K-3 mendapatkan akurasi 76%

Catatan : Nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data.

Tabel 4. 26 Hasil data uji k=5

Kelas	Matang	Mentah
Matang	12	3
Mentah	3	12

Jumlah data yang diklasifikasikan benar (kotak kuning) = 24, jumlah data yang diklasifikasikan salah (kotak hijau) = 6.

Keterangan:

(TP) *True Positive* / Merupakan buah nanas matang yang terdeteksi matang oleh sistem = 12

(TN) *True Negative* / Merupakan buah nanas mentah yang terdeteksi mentah oleh sistem = 12

(FP) *False Positive* / Merupakan buah nanas mentah yang dideteksi matang oleh sistem = 3

(FN) *False Negative* / Merupakan buah nanas matang yang dideteksi mentah oleh sistem = 3

$$Akurasi = \frac{TP + TN}{TP + FP + FN + TN} \times 100\%$$

$$Akurasi = \frac{24}{30} \times 100\% = 80\%$$

Jadi, hasil data uji K-5 mendapatkan akurasi 80%

Catatan : Nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data.

Tabel 4. 27 Hasil data uji k=7

Kelas	Matang	Mentah
Matang	11	4
Mentah	2	13

Jumlah data yang diklasifikasikan benar (kotak kuning) = 24, jumlah data yang diklasifikasikan salah (kotak hijau) = 6.

Keterangan:

(TP) *True Positive* / Merupakan buah nanas matang yang terdeteksi matang oleh sistem = 11

(TN) *True Negative* / Merupakan buah nanas mentah yang terdeteksi mentah oleh sistem = 13

(FP) *False Positive* / Merupakan buah nanas mentah yang dideteksi matang oleh sistem = 2

(FN) *False Negative* / Merupakan buah nanas matang yang dideteksi mentah oleh sistem = 4

$$Akurasi = \frac{TP + TN}{TP + FP + FN + TN} \times 100\%$$

$$Akurasi = \frac{24}{30} \times 100\% = 80\%$$

Jadi, hasil data uji K-7 mendapatkan akurasi 80%

Catatan : Nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data.

Tabel 4. 28 Hasil data uji k=9

Kelas	Matang	Mentah
Matang	11	4
Mentah	2	13

Jumlah data yang diklasifikasikan benar (kotak kuning) = 24, jumlah data yang diklasifikasikan salah (kotak hijau) = 6.

Keterangan:

(TP) *True Positive* / Merupakan buah nanas matang yang terdeteksi matang oleh sistem = 11

(TN) *True Negative* / Merupakan buah nanas mentah yang terdeteksi mentah oleh sistem = 13

(FP) *False Positive* / Merupakan buah nanas mentah yang dideteksi matang oleh sistem = 2

(FN) *False Negative* / Merupakan buah nanas matang yang dideteksi mentah oleh sistem = 4

$$Akurasi = \frac{TP + TN}{TP + FP + FN + TN} \times 100\%$$

$$Akurasi = \frac{24}{30} \times 100\% = 80\%$$

Jadi, hasil data uji K-9 mendapatkan akurasi 80%

Catatan : Nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data.

4.2.6 *System Requirements*

1. CPU: 2 x 64-bit 2.8 GHz 8.00 GT/s CPUs
2. RAM: 32 GB (or 16 GB of 1600 MHz DDR3 RAM)
3. Storage: 300 GB. (600 GB for air-gapped deployments.) Additional space recommended if the repository will be used to store packages built by the customer. With an empty repository, a base install requires 2 GB.
4. Internet access to download the files from Anaconda.org or a USB drive containing all of the files you need with alternate instructions for air gapped installations.

BAB V

PEMBAHASAN

5.1 Pembahasan Model

Data yang digunakan dalam penelitian ini terdiri dari data real peneliti yang di dapat dengan mengambil gambar secara langsung dengan menggunakan kamera smarthphone VIVO Z1 Pro dengan resolusi kamera 16 MP. Masing masing 209 data training dan 30 data testing sehingga total dataset yang di gunakan pada penelitian ini yaitu 239 data set.

Berdasarkan hasil penelitian data yang di ambil terdiri dari 2 kategori yaitu Matang dan Mentah. Selanjutnya citra buah nanas di ekstrasi ciri menggunakan *Gray Level Co-occurrence Matrix*. Setelah di dapatkan ciri dari citra buah nanas selanjutnya dilakukan proses klasifikasi menggunakan *K-Nearest Neighbor*. Pada tahap ini citra pada buah nanas dijadikan sebagai inputan untuk mencari nilai akurasi terbaik pada percobaan model K-NN.

Pada percobaan model K-NN dengan jumlah nilai K-1, menggunakan arah GLCM 0^0 dengan jarak 1, mendapatkan nilai akurasi sebesar 80% dengan 30 data testing dan 209 data training, dengan kategori matang dan mentah. Selanjutnya percobaan model K-NN dengan jumlah nilai K-3 dengan arah GLCM 0^0 dan jarak 1 mendapatkan nilai akurasi 76% dan Percobaan menggunakan model K-NN dengan jumlah K-5, K=7 dan K=9 dengan arah GLCM 0^0 dan jarak 1 mendapatkan nilai akurasi yang sama yaitu 80%.

Percobaan selanjutnya menggunakan arah GLCM = 45^0 dengan jarak = 1, pada percobaan model K-NN dengan jumlah nilai K-1 mendapatkan nilai akurasi sebesar 80%, dan percobaan model K-NN dengan jumlah nilai K-3 dan K-5 dengan GLCM = 45^0 dengan jarak = 1 mendapatkan nilai akurasi yang sama yakni 76%.

Pada percobaan terakhir menggunakan arah GLCM = 90^0 dengan jarak =1, pada model K-NN dengan jumlah nilai K-7 dan K-9 juga mendapatkan nilai akurasi yang sama yaitu sebesar 83%.

Pada tahap Confusion Matrix terdapat kendala pengenalan tingkat kematangan, beberapa data keluar dengan hasil yang tidak sesuai. Arah dan jarak cukup berpengaruh untuk mendapatkan hasil yang sesuai. Berdasarkan hasil perhitungan diatas percobaan dengan arah 90^0 dengan jarak 1 mendapatkan akurasi tertinggi yaitu sebesar 83%. Pada penelitian ini disimpulkan bahwa arah derajat tertinggi memperoleh nilai akurasi yang lebih baik. dapatkan dari pengujian metode GLCM dan K-NN, dengan arah 90^0 dan jarak 1 mendapatkan nilai akurasi tertinggi yaitu 83% pada percobaan model K-NN K-7 dan K-9. Pada penelitian ini penggunaan metode K-NN sudah cukup baik untuk tahap klasifikasi.

Tabel 5. 1 Hasil Pembahasan Model

Nilai K	Arah	Jarak	Hasil Akurasi
K=1	0^0	1	80%
K=3	0^0	1	76%
K=5	0^0	1	80%
K=7	0^0	1	80%
K=9	0^0	1	80%
K=1	45^0	1	80%
K=3	45^0	1	76%
K=5	45^0	1	76%
K=7	90^0	1	83%
K=9	90^0	1	83%

Berdasarkan hasil perhitungan diatas, deteksi kematangan buah nanas yang diukur menggunakan *confusion matrix* memperoleh nilai akurasi tertinggi sebesar 83% pada nilai k=7 dan k=9 dengan arah 90^0 dan jarak 1.

5.2 Pembahasan Sistem

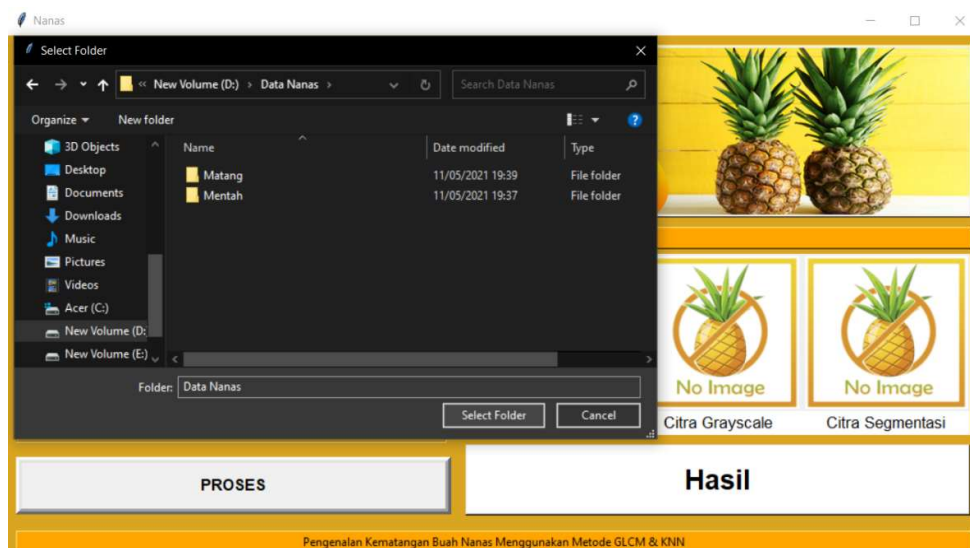
1) Tampilan Menu Utama

Tampilan menu utama merupakan form utama yang muncul saat program dijalankan. Pada form menu utama menampilkan menu – menu yang ada pada sistem yaitu menu load data training, menu load data testing, dan proses.



Gambar 5. 1 Tampilan Menu Utama

2) Tampilan Proses Load Data Training (Proses pengambilan data training)



Gambar 5. 2 Tampilan Proses Load Data Training

3) Tampilan Proses Load Data Testing (Proses pengambilan data testing)



Gambar 5. 3 Tampilan Proses Load Data Testing

4) Tampilan Hasil Pengenalan



Gambar 5. 4 Tampilan Hasil Pengenalan

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan dan pembahasan yang telah diuraikan sebelumnya maka dapat ditarik suatu kesimpulan bahwa:

Hasil penerapan klasifikasi K-NN (*K-Nearest Neighbor*) dan ekstraksi fitur GLCM (*Gray Level Co-occurrence Matrix*) untuk pengenalan kematangan buah nanas, yang diukur dengan *confusion matrix* memperoleh nilai akurasi paling tinggi sebesar 83% pada nilai K=9 dan K=7 dengan GLCM arah 0⁰ dan jarak 1, dengan hasil buah matang yang teridentifikasi benar berjumlah 12, dan buah matang yang teridentifikasi salah berjumlah 3 buah. Sedangkan buah mentah yang teridentifikasi benar berjumlah 13, dan yang mentah teridentifikasi salah berjumlah 2 buah.

6.2 Saran

Dari penerapan metode *Gray Level Co-occurrence Matrix* dan *K-Nearest Neighbor* untuk pengenalan tingkat kematangan buah nanas, ada beberapa yang perlu diperhatikan:

1. Penulis berharap untuk penelitian selanjutnya agar memperoleh nilai akurasi yang lebih baik.
2. Untuk pemrosesan yang lebih cepat disarankan agar menggunakan citra yang sudah dikompresi agar ukuran gambar tidak terlalu besar .
3. Untuk pengambilan citra disarankan agar memiliki pencahayaan yang baik agar mendapatkan hasil / akurasi yang lebih baik

DAFTAR PUSTAKA

- [1] A. Lustini, Saparudin, and I. Primanita, “KLASIFIKASI TINGKAT KEMATANGAN BUAH NANAS MENGGUNAKAN RUANG WARNA RED – GREEN – BLUE DAN HUE – SATURATION – INTENSITY,” *Digit. Teknol. Inf.*, vol. 2, no. 1, pp. 1–8, 2019.
- [2] M. Widyaningsih, “IDENTIFIKASI KEMATANGAN BUAH APEL DENGAN GRAY LEVEL CO - OCCURRENCE MATRIX (GLCM),” vol. 6, no. 1, pp. 71–88, 2016.
- [3] Indarto and Murinto, “Deteksi Kematangan Buah Pisang Berdasarkan Fitur Warna Citra Kulit Pisang Menggunakan Metode Transformasi Ruang Warna HIS (Banana Fruit Detection Based on Banana Skin Image Features Using HSI Color Space Transformation Method),” vol. V, no. 1, pp. 15–21, 2017.
- [4] R. Widodo, A. W. Widodo, and A. Supriyanto, “Pemanfaatan Ciri Gray Level Co-Occurrence Matrix (GLCM) Citra Buah Jeruk Keprok (Citrus reticulata Blanco) untuk Klasifikasi Mutu,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 11, pp. 5769–5776, 2018.
- [5] A. Pariyandani, “Klasifikasi Citra Ikan Berformalin Menggunakan Metode K-NN dan GLCM,” *Pros. Semin. Nas. Teknol. Inform.*, vol. 2, no. 1, pp. 42–47, 2019.
- [6] C. Iswahyudi, “Prototype Aplikasi Untuk Mengukur Kematangan Buah Apel Berdasarkan Kemiripan Warna,” *J. Teknol.*, vol. 3, no. 2, pp. 107–112, 2010.
- [7] I. A. Halela, B. Nurhadiyono, S. Si, M. Kom, and F. Z. Rahmanti, “Identifikasi Jenis Buah Apel Menggunakan Algoritma K-Nearest Neighbor (KNN) dengan Ekstraksi Fitur Histogram,” pp. 1–8.
- [8] L. Anshori, R. R. M. Putri, and Tibyani, “Implementasi Metode K-Nearest

Neighbor untuk Rekomendasi Keminatan Studi (Studi Kasus: Jurusan Teknik Informatika Universitas Brawijaya),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 7, pp. 2745–2753, 2018, [Online]. Available: https://www.researchgate.net/publication/321097020_Implementasi_Metode_K-Nearest_Neighbor_untuk_Rekomendasi_Keminatan_Studi_Studi_Kasus_Jurusan_Teknik_Informatika_Universitas_Brawijaya.

- [9] Arwansyah and J. s. Arie, “Implementasi Algoritma KNN Dalam Memprediksi Curah Hujan dan Temperatur Untuk Tanaman Padi,” *Pros. Semin. Ilm. Sist. Inf. DAN Teknol. Inf.*, vol. 8, no. 1, pp. 11–20, 2019.
- [10] A. Risvana, M. K. Adi, Prajanto Wahyu, and M. K. Astuti, Setia, S.Si, “KLASIFIKASI TINGKAT KEMATANGAN BUAH JAMBU BIJI MERAH DENGAN ALGORITMA K-NEAREST NEIGHBOR,” pp. 1–9, 2012.
- [11] Kaharuddin, Kusri, and E. T. Luthfi, “KLASIFIKASI JENIS REMPAH-REMPAH BERDASARKAN FITUR WARNA RGB DAN TEKSTUR MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOR,” *J. Inf. Interaktif*, vol. 4, no. 1, pp. 17–22, 2019.
- [12] R. Munarto, E. Permata, and R. Salsabilla, “Klasifikasi Kualitas Biji Jagung Manis Berdasarkan Fitur Warna Menggunakan Fuzzy Logic,” *Simp. Nas. RAPI XIII - 2014 FT UMS*, pp. 5–12, 2014.
- [13] P. Mayadewi and E. Rosely, “Prediksi Nilai Proyek Mahasiswa Menggunakan Algoritma Klasifikasi Data Mining,” *Semin. Nas. Sist. Inf. Indones.*, no. November, pp. 2–3, 2015.
- [14] M. F. Rahman, D. Alamsah, and M. I. Darmawidjadja, “Klasifikasi Untuk Diagnosa Diabetes Menggunakan Metode Bayesian Regularization Neural Network (RBNN),” *J. Inform.*, vol. 11, no. 1, pp. 36–45, 2017, doi: 10.26555/jifo.v11i1.a5452

LAMPIRAN

Lampiran 1. 1 Kode Program

1. PEMBUATAN FORM

```
import tkinter as tk
import tkinter.filedialog as fdialog
import tkinter.ttk as ttk
from PIL import Image, ImageTk
import GLCMKNN as lt1

class nanas(tk.Frame):
    def __init__(self, parent):
        self.parent = parent
        self.frame()
        self.widget()
        self.hasil()

    def frame(self):
        #frame judul
        self.frame1 = tk.Frame(self.parent, width=1000, height=180,
borderwidth=1, background="#FFD700", relief='raised')
        self.frame1.grid(row=0, column=0, padx=10, pady=10, columnspan=2,
sticky="NSEW")
        self.frame1.grid_propagate(0)
        #frame Label input data
        self.frame4 = tk.Frame(self.parent, width=1000, height=30,
borderwidth=1, background="#FFA500", relief='raised')
        self.frame4.grid(row=1, column=0, padx=10, columnspan=2,
sticky="NSEW")
```

```

        self.frame4.grid_propagate(0)
#frame input
        self.frame2 = tk.Frame(self.parent, width=450, height=294,
borderwidth=0, padx=10, pady=5, background="#DAA520", relief='raised')
        self.frame2.grid(row=2, column=0, columnspan=2, sticky="NSEW")
        self.frame2.grid_propagate(0)
#frame gambar
        self.frame3 = tk.Frame(self.parent, width=550, height=294,
borderwidth=0, padx=10, pady=5, background="#DAA520", relief='raised')
        self.frame3.grid(row=2, column=1, columnspan=2, sticky="NSEW")
        self.frame3.grid_propagate(0)
#frame credit
        self.frame5 = tk.Frame(self.parent, width=1000, height=30,
borderwidth=1, background="#FFA500", relief='raised')
        self.frame5.grid(row=4, column=0, padx=10, columnspan=2,
sticky="NSEW")
        self.frame5.grid_propagate(0)

    def widget(self):
#frame 2
=====

#data training
#label
        self.labeldatatrain = tk.Label(self.frame2, text='Data Latih', relief='ridge',
width=12, bg="white")
        self.labeldatatrain.grid(row=1, column=0, sticky="W")

#input
        value_datatraining = tk.StringVar(self.frame2, value="")
        self.inputdatatrain = tk.Entry(self.frame2,
textvariable=value_datatraining, relief='ridge', width=41)

```

```

self.inputdatatrain.grid(row=1, column=1, padx=3, pady=3, sticky="W")

#button
self.buttondatatrain = tk.Button(self.frame2, text='Pilih', width=13,
command=lambda:value_datatraining.set(fdialog.askdirectory()))
self.buttondatatrain.grid(row=1, column=2, padx=3, pady=3, sticky="W")

#=====

#data testing
#label
self.labeldatatest = tk.Label(self.frame2, text='Data Uji', relief='ridge',
width=12, bg="white")
self.labeldatatest.grid(row=2, column=0, sticky="W")
#input
value_datatesting = tk.StringVar(self.frame2, value="")
self.inputdatatest = tk.Entry(self.frame2, textvariable=value_datatesting,
relief='ridge', width=41)
self.inputdatatest.grid(row=2, column=1, padx=3, pady=3, sticky="W")
#button
self.buttondatatesting = tk.Button(self.frame2, text='Pilih', width=13,
command=lambda:value_datatesting.set(fdialog.askopenfilename()))
self.buttondatatesting.grid(row=2, column=2, padx=3, pady=3,
sticky="W")
#frame3
#=====

# GLCM
#label
self.labelarah = tk.Label(self.frame2, text='Arah', relief='ridge', width=12,
bg="white")
self.labelarah.grid(row=4, column=0, pady=5, sticky="W")
#combobox

```

```

self.valdist = tk.StringVar(self.frame2, value="")
self.cboxarah = ttk.Combobox(self.frame2, textvariable=self.valdist,
width=56, state='readonly')
self.cboxarah['values']=('0','45','90','135')
self.cboxarah.current(0)
self.cboxarah.grid(row=4, column=1, columnspan=2, padx=3, pady=3,
sticky="W")
#label
self.labeljarak = tk.Label(self.frame2, text='Jarak', relief='ridge',
width=12, bg="white")
self.labeljarak.grid(row=5, column=0, pady=5, sticky="W")
#combobox2
self.valdist = tk.StringVar(self.frame2, value="")
self.cboxjarak = ttk.Combobox(self.frame2, textvariable=self.valdist,
width=56, state='readonly')
self.cboxjarak['values']=('1','2','3','4')
self.cboxjarak.current(0)
self.cboxjarak.grid(row=5, column=1, columnspan=2, padx=3, pady=3,
sticky="W")
#buttonproses
self.buttonproses = tk.Button(self.frame2, text='PROSES', font="Arial 12
bold", width=35, height=2, command=self.proses)
self.buttonproses.grid(row=7, column=0, columnspan=3, pady=10,
sticky="NSEW")
self.buttonproses.configure(borderwidth=6, relief="raised",
anchor="center")

#grupbox
self.grupboxinput = tk.LabelFrame(self.frame2, height=20, width=450,
bg="#DAA520", text="Silahkan Masukkan Data..", fg="black")

```

```
self.grupboxinput.grid(row=0, column=0, columnspan=3, pady=3,  
sticky="W")
```

```
self.grupboxinput.grid_propagate(0)
```

```
self.glcm = tk.LabelFrame(self.frame2, height=20, width=450,  
bg="#DAA520", text="Gray Level Co-Occurrence Matrix", fg="black")
```

```
self.glcm.grid(row=3, column=0, columnspan=3, sticky="W")
```

```
self.glcm.grid_propagate(0)
```

```
self.line = tk.LabelFrame(self.frame2, height=20, width=450,  
bg="#DAA520", text="", fg="black")
```

```
self.line.grid(row=6, column=0, pady=5, columnspan=3, sticky="W")
```

```
self.line.grid_propagate(0)
```

```
#frame4
```

```
=====
```

```
#label input data
```

```
self.labeltext1 = tk.Label(self.frame4, text="PROSES EKSTRAKSI DAN  
KLASIFIKASI", background="#FFA500", anchor="center")
```

```
self.labeltext1.grid()
```

```
self.labeltext1.pack()
```

```
#frame5
```

```
=====
```

```
#credit
```

```
self.labeltext2 = tk.Label(self.frame5, text="Pengenal Kematangan  
Buah Nanas Menggunakan Metode GLCM & KNN", background="#FFA500",  
anchor="center")
```

```
self.labeltext2.grid()
```

```
self.labeltext2.pack()
```

```

#panel
gambar=====

def hasil(self):

    img = Image.open("nomageasli.jpg")
    img = img.resize((170, 160), Image.ANTIALIAS)
    img = ImageTk.PhotoImage(img)

    self.panel1 = tk.Label(self.frame3, image=img)
    self.panel1.image_names = img
    self.panel1.grid(row=1, column=0, rowspan=5, sticky="NSEW")
    self.labelpanel1 = tk.Label(self.frame3, text='Citra Asli', font="arial 13",
bg="white")
    self.labelpanel1.grid(row=9, rowspan=2, column=0, sticky="NSEW")

    self.panel2 = tk.Label(self.frame3, image=img)
    self.panel2.image_names = img
    self.panel2.grid(row=1, column=1, rowspan=5, sticky="NSEW")
    self.labelpanel2 = tk.Label(self.frame3, text='Citra Grayscale', font="arial
13", bg="white")
    self.labelpanel2.grid(row=9, rowspan=2, column=1, sticky="NSEW")

    self.panel3 = tk.Label(self.frame3, image=img)
    self.panel3.image_names = img
    self.panel3.grid(row=1, column=2, rowspan=5, sticky="NSEW")
    self.labelpanel3 = tk.Label(self.frame3, text='Citra Segmentasi',
font="arial 13", bg="white")
    self.labelpanel3.grid(row=9, rowspan=2, column=2, sticky="NSEW")

#Hasil

```

```
self.labelhasil = tk.Label(self.frame3, text="Hasil", font="arial 22 bold",
height=2, width=29, background="white", relief="raised")
self.labelhasil.grid(row=11, column=0, columnspan=3, sticky="W",
pady=10)
```

```
banner = Image.open("banerasli.jpg")
banner = banner.resize((994, 174), Image.ANTIALIAS)
banner = ImageTk.PhotoImage(banner)
```

```
self.panel4 = tk.Label(self.frame1, image=banner)
self.panel4.image_names = banner
self.panel4.grid(row=0, column=0, rowspan=5, sticky="NSEW")
```

```
#=====
```

```
def proses(self):
    datatraining = self.inputdatatrain.get()
    datatesting = self.inputdatatest.get()
    arah = self.cboxarah.get()
    jarak = self.cboxjarak.get()
    a = lt1.imageprocessing()
    result, grayscale, hasil_segment = a.predicts(datatesting, datatraining,
jarak, arah)
    print(result)
```

```
gambarasli = Image.open(datatesting)
gambarasli = gambarasli.resize((170, 160), Image.ANTIALIAS)
gambarasli = ImageTk.PhotoImage(gambarasli)
self.panel1.configure(image = gambarasli)
self.panel1.image_name = gambarasli
```

```

gambargray = Image.fromarray( grayscale )
gambargray = gambargray.resize((170, 160), Image.ANTIALIAS)
gambargray = ImageTk.PhotoImage(gambargray)
self.panel2.configure(image = gambargray)
self.panel2.image_name = gambargray

gambarsegment = Image.fromarray(hasil_segment)
gambarsegment = gambarsegment.resize((170, 160), Image.ANTIALIAS)
gambarsegment = ImageTk.PhotoImage(gambarsegment)
self.panel3.configure(image = gambarsegment)
self.panel3.image_name = gambarsegment

self.labelhasil.config(text=result[0])

```

```

#=====

```

```

def main():
    root = tk.Tk()
    root.geometry('1021x545+50+50')
    root.configure(background='#DAA520')
    app = nanas(root)
    app.parent.title("Nanas")
    root.mainloop()
if __name__ == '__main__':
    main()

```

2. Image Processing

```
import cv2
from imutils import paths
import os
import numpy as np
from skimage import feature
from sklearn import neighbors
from matplotlib import pyplot as plt

class imageprocessing:

    def imPaths(self, imfolder):
        patharray = []
        for item in paths.list_images(imfolder):
            patharray.append(item)
        return patharray

    def preprocessing_1(self, img):
        grayscale = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        grayscale = cv2.resize(grayscale, (150, 150),
                                interpolation=cv2.INTER_AREA)
        equalize = cv2.equalizeHist(grayscale)
        blur = cv2.GaussianBlur(equalize, (5,5), 0)
        ret, alpha = cv2.threshold(np.uint8(blur), 50,255,
                                    cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)

        alpha[alpha==255] = 1
```

```

hasil_segment = alpha * equalize
cnts = cv2.findContours(hasil_segment, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
cnts = cnts[0] if len(cnts) == 2 else cnts[1]
cnts = sorted(cnts, key=cv2.contourArea, reverse=True)
for c in cnts:
    x,y,w,h = cv2.boundingRect(c)
    ROI = hasil_segment[y:y+h, x:x+w]
    break
return ROI, grayscale

```

```

def ekstraksi_fitur(self, img, distance, angle):
    glcm = feature.greycomatrix(img, [1], [0],
symmetric='false',normed='true')

```

```

    con = feature.greycoprops(glcm, 'contrast')[0,0]
    energy = feature.greycoprops(glcm, 'energy')[0,0]
    hom = feature.greycoprops(glcm, 'homogeneity')[0,0]
    return [con,energy,hom]

```

```

def getClass(self, imfolder, distance, angle):
    label = []
    impath = self.imPaths(imfolder)
    for item in impath:
        label.append(os.path.split(os.path.dirname(item))[-1])
    return label

```

```

def create_datalatih(self, imfolder, distance, angle):
    data_latih = []
    impath = self.imPaths(imfolder)
    for item in impath:

```

```

        img = cv2.imread(item)
        hasil_preprocessing, grayscale = self.preprocessing_1(img)
        hasil_ekstraksi = self.ekstraksi_fitur(hasil_preprocessing, distance,
angle)
        data_latih.append(hasil_ekstraksi)
    return data_latih

def create_datatesting(self, pathimg, distance, angle):
    data_testing = []
    img = cv2.imread(pathimg)
    hasil_preprocessing, grayscale = self.preprocessing_1(img)
    hasil_ekstraksi = self.ekstraksi_fitur(hasil_preprocessing, distance, angle)
    data_testing.append(hasil_ekstraksi)
    return data_testing, grayscale, hasil_preprocessing

def model_knn(self, imfolder, distance, angle):
    x = self.create_datalatih(imfolder, distance, angle)
    y = self.getClass(imfolder, distance, angle)
    clf = neighbors.KNeighborsClassifier(n_neighbors=3, weights='distance',
algorithm='auto', metric='euclidean')
    clf.fit(x,y)
    return clf

def predicts(self, imtest, imfolder, distance, angle):
    data_test, grayscale, hasil_segment = self.create_datatesting(imtest,
distance, angle)
    model = self.model_knn(imfolder, distance, angle)
    hasil_klasifikasi = model.predict(data_test)
    return hasil_klasifikasi, grayscale, hasil_segment

```

Lampiran 1. 2 RIWAYAT HIDUP PENELITI



Nama : Mohamad Rizal Kasim
NIM : T3116067
Tempat, Tanggal Lahir : Hunggaluwa, 04 April 1998
Email : Mamatkasim50@gmail.com

Riwayat Pendidikan :

1. Tahun 2010, menyelesaikan Pendidikan di Sekolah Dasar Negeri 1 Ilangata, sekarang SDN 4 Ilangata, Kecamatan Anggrek Kabupaten Gorontalo
2. Tahun 2013, menyelesaikan Pendidikan di Sekolah Madrasah Tsanawiyah Negeri Model Limboto, Kecamatan Limboto Kabupaten Gorontalo
3. Tahun 2016, menyelesaikan Pendidikan di Sekolah Menengah Kejuruan Negeri 3 Gorontalo, Kecamatan Kota Tengah, Kota Gorontalo
4. Tahun 2016, telah diterima menjadi Mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo.

Lampiran 1. 3 Surat Balasan Penelitian



PEMERINTAH KABUPATEN BOLAANG MONGONDOW
KECAMATAN PASSI BARAT
DESA LOBONG

SURAT KETERANGAN

NOMOR : 53/SK/DL/KPB/IV/2021

Sangadi Desa Lobong menerangkan dengan benar kepada :

N a m a : **Mohamad Rizal Kasim**
Tempat /tgl lahir : Hunggaluwa, 04-04-1998
Nim : T3116067
Jenis Kelamin : Laki-laki.
A g a m a : Islam
Pekerjaan : Mahasiswa.
Fakultas : Fakultas Ilmu Komputer.
Program studi : Teknik Informatika
Judul penelitian : PENGENALAN KEMATANGAN BUAH NANAS
MENGGUNAKAN METODE GRAY LEVEL CO-
OCCURENCE MATRIX (GLCM) DAN K-NEAREST
NEIGHBORS (K-NN).

Bahwa nama yang bersangkutan diatas benar telah melaksanakan pengambilan data dalam rangka penyusunan **Proposal atau Skripsi**.

Demikian Surat Keterangan ini dibuat dengan benar untuk keperluan Dan digunakan sebagaimana mestinya.

Lobong, 13 april 2021

SANGADI LOBONG



RIFAI MOKOGINTA

Lampiran 1. 4 Surat Rekomendasi Bebas Plagiasi



**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS ICHSAN
(UNISAN) GORONTALO**

SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829978 Gorontalo

SURAT REKOMENDASI BEBAS PLAGIASI

No. 0912/UNISAN-G/S-BP/VI/2021

Yang bertanda tangan di bawah ini :

Nama : Sunarto Taliki, M.Kom
NIDN : 0906058301
Unit Kerja : Pustikom, Universitas Ichsan Gorontalo

Dengan ini Menyatakan bahwa :

Nama Mahasisw : MOHAMAD RIZAL KASIM
NIM : T3116067
Program Studi : Teknik Informatika (S1)
Fakultas : Fakultas Ilmu Komputer
Judul Skripsi : PENGENALAN KEMATANGAN BUAH NANAS
MENGUNAKAN METODE GRAY LEVEL
CO-OCCURRENCE MATRIX (GLCM) DAN
K-NEAREST NEIGHBOR (K-NN)

Sesuai dengan hasil pengecekan tingkat kemiripan skripsi melalui aplikasi Turnitin untuk judul skripsi di atas diperoleh hasil Similarity sebesar 20%, berdasarkan SK Rektor No. 237/UNISAN-G/SK/IX/2019 tentang Panduan Pencegahan dan Penanggulangan Plagiarisme, bahwa batas kemiripan skripsi maksimal 35% dan sesuai dengan Surat Pernyataan dari kedua Pembimbing yang bersangkutan menyatakan bahwa isi softcopy skripsi yang diolah di Turnitin SAMA ISINYA dengan Skripsi Aslinya serta format penulisannya sudah sesuai dengan Buku Panduan Penulisan Skripsi, untuk itu skripsi tersebut di atas dinyatakan BEBAS PLAGIASI dan layak untuk diujikan.

Demikian surat rekomendasi ini dibuat untuk digunakan sebagaimana mestinya.

Gorontalo, 21 Juni 2021
Tim Verifikasi,



Sunarto Taliki, M.Kom
NIDN. 0906058301

Tembusan :

1. Dekan
2. Ketua Program Studi
3. Pembimbing I dan Pembimbing II
4. Yang bersangkutan
5. Arsip

T3116067 MOHAMAD RIZAL KASIM

PENGENALAN KEMATANGAN BUAH NANAS MENGGUNAKAN ...

Sources Overview

20%

OVERALL SIMILARITY

1	e-journal.janabdra.ac.id	4%
2	ionpcd.blogspot.com	2%
3	Zulfrianto Y Lamasigi, Maryam Hasan, Yulianti Lasena. "Local Binary Pattern untuk Pengenalan Jenis Daun Tanaman Obat menggunakan ...	2%
4	openlibrarypublications.telkomuniversity.ac.id	1%
5	j-ptik.ub.ac.id	1%
6	docs.anaconda.com	1%
7	docplayer.info	<1%
8	www.scribd.com	<1%
9	jurnal.usbykp.ac.id	<1%
10	eprints.uns.ac.id	<1%
11	begawe.unram.ac.id	<1%
12	repositor.umm.ac.id	<1%
13	text-id.123dok.com	<1%
14	eprints.undip.ac.id	<1%
15	repository.uin-suska.ac.id	<1%
16	prosiding.senadi.upy.ac.id	<1%

17	ejurnal.ung.ac.id	INTERNET	<1%
18	stmikplk.ac.id	INTERNET	<1%
19	ocs.unud.ac.id	INTERNET	<1%
20	www.icg.isy.liu.se	INTERNET	<1%
21	jurnal.untan.ac.id	INTERNET	<1%
22	semantika.polgan.ac.id	INTERNET	<1%
23	Budy Santoso, Azminuddin I. S. Azis, Andi Bode. "Pengendalian Lampu Lalu Lintas Cerdas di Persimpangan Empat Ruas yang Komple...	CROSSREF	<1%
24	achenamono.blogspot.com	INTERNET	<1%
25	digilib.unhas.ac.id	INTERNET	<1%
26	journal.lirpi.or.id	INTERNET	<1%

Excluded search repositories:

- Submitted Works

Excluded from Similarity Report:

- Small Matches (less than 25 words).

Excluded sources:

- None