

**DETEKSI KESEGARAN IKAN LAYANG
BERDASARKAN CITRA DAGING
MENGUNAKAN METODE
GLCM DAN KNN**

Oleh

SITI REGINA KHAIRUNNIZA A. AHMAD

T3116064

SKRIPSI

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



**PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2023**

HALAMAN JUDUL

**DETEKSI KESEGARAN IKAN LAYANG
BERDASARKAN CITRA DAGING
MENGUNAKAN METODE
GLCM DAN KNN**

Oleh

SITI REGINA KHAIRUNNIZA A. AHMAD

T3116064

SKRIPSI

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



**PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2023**

PENGESAHAN SKRIPSI

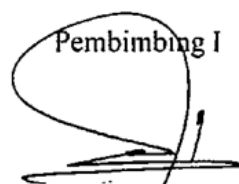
DETEKSI KESEGARAN IKAN LAYANG BERDASARKAN CITRA DAGING MENGUNAKAN METODE GLCM DAN KNN

Oleh

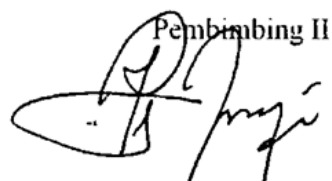
SITI REGINA KHAIRUNNIZA A. AHMAD
T3116064

SKRIPSI

Untuk Memenuhi Syarat Ujian Guna Memperoleh
Gelar Sarjana Program Studi Teknik Infomatika
Ini Telah disetujui dan siap untuk diseminarkan
Gorontalo, 17 Mei 2023

Pembimbing I


Sunarto Taliki M.Kom
NIDN : 0906058301

Pembimbing II


Kartika Chandra Pelangi, M.Kom
NIDN : 0916038304

PERSETUJUAN SKRIPSI

DETEKSI KESEGARAN IKAN LAYANG BERDASARKAN CITRA DAGING MENGUNAKAN METODE GLCM DAN KNN

Oleh

SITI REGINA KHAIRUNNIZA A. AHMAD

T3116064

Diperiksa Oleh Panitia Ujian Stars Satu (S1)
Universitas Ichsan Gorontalo
Gorontalo, Mei 2023

1. Ketua Penguji
Asmaul Husnah Nasrullah, M.Kom
2. Anggota
Husdi, M.Kom
3. Anggota
Muis Nanja, M.Kom
4. Anggota
Sunarto Taliki, M.Kom
5. Anggota
Kartika Chandra Pelangi, M.Kom



Dekan Fakultas Ilmu Komputer



Irvan Abraham Salihi, S.Kom., M.Kom
NIDN. 0912117702

Ketua Program Studi



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, Mei 2023

Yang Membuat Pernyataan



SITI REGINA K.A. AHMAD

ABSTRACT

SITI REGINA KHAIRUNNIZA A AHMAD. T3116064. MACKEREL SCADS FRESHNESS DETECTION BASED ON MEAT IMAGE USING GLCM AND KNN METHODS.

Mackerel scads is a type of marine fish that is often consumed by the Gorontalo people because of its affordable price and has a high protein content. It contains nutrients and protein that are very good for the human body. In selecting fresh fish so far, it is still using a manual process based on the human's sense of sight and observation. In this case, a system is needed that can detect the freshness of mackerel scads automatically to facilitate humans in the process of selecting fresh fish. Therefore, in this study, a system is created to classify freshness based on the level of accuracy using the extraction feature method, namely Gray Level Co-occurrence Matrix (GLCM). In this study, 259 training data are employed, and each consists of 140 fresh and 119 not fresh images. The image data is processed into a grayscale image which is then detected using color texture. After obtaining the quality of mackerel scads, the calculation of characteristics using Gray Level Co-occurrence Matrix (GLCM) is then carried out. The algorithm used for the classification process is the K-Nearest Neighbor (K-NN) algorithm. The application of Gray Level Co-occurrence Matrix (GLCM) and K-Nearest Neighbor (K-NN) methods for freshness detection of mackerel scads with the number of $K = 5$, GLCM direction = 00 and distance = 5 gets an accuracy of 85.5% by calculating using the Confusion Matrix formula.

Keywords: mackerel scads, detection, GLCM, KNN



ABSTRAK

SITI REGINA KHAIRUNNIZA A AHMAD. T3116064. DETEKSI KESEGERAN IKAN LAYANG BERDASARKAN CITRA DAGING MENGGUNAKAN METODE GLCM DAN KNN

Ikan layang adalah jenis ikan laut yang sering dikonsumsi oleh masyarakat Gorontalo karena harganya yang terjangkau dan memiliki kandungan protein tinggi serta mengandung gizi dan protein yang sangat baik bagi tubuh manusia. Dalam pemilihan ikan yang segar selama ini masih menggunakan proses manual dengan berdasarkan indra penglihatan dan pengamatan manusia. Dalam hal ini dibutuhkan suatu sistem yang dapat mendeteksi kesegaran ikan layang secara otomatis sehingga mempermudah manusia dalam proses pemilihan ikan segar. Oleh karena itu dalam penelitian ini akan dibuat sistem untuk mengklasifikasikan kesegaran berdasarkan tingkat akurasi dengan menggunakan metode fitur ekstraksi yaitu *GrayLevel Co-occurrence Matrix* (GLCM). Dalam penelitian ini digunakan data latih sebanyak 259 citra daging ikan layang yang masing-masing terdiri dari segar 140 dan tidak segar 119. Data citra tersebut diolah menjadi citra grayscale yang kemudian dilakukan deteksi kualitas ikan layang menggunakan tekstur warna. Setelah didapat kualitas ikan layang kemudian dilakukan perhitungan ciri menggunakan *Gray Level Co-occurrence Matrix* (GLCM). Algoritma yang digunakan untuk proses klasifikasi adalah algoritma *K-Nearest Neighbor* (K-NN). Penerapan metode *Gray Level Co-occurrence Matrix* (GLCM) dan *K-Nearest Neighbor* (K-NN) untuk deteksi kesegaran ikan layang dengan jumlah $K=5$, arah GLCM = 0^0 dan jarak = 5 mendapatkan hasil akurasi 85,5% dengan melakukan perhitungan menggunakan rumus Confusion Matriks.

Kata kunci : ikan layang, deteksi, GLCM, KNN



KATA PENGANTAR

Assalamu'alaikum Wr.Wb

Puji syukur penulis panjatkan kehadirat Allah SWT atas segala nikmat dan hidayah-Nya. Shalawat serta salam tak lupa penulis haturkan kepada Nabi Besar Muhammad SAW beserta para keluarga, sahabat dan para pengikutnya, sehingga penulis dapat menyelesaikan usulan penelitian ini dengan judul “DETEKSI KESEGERAN IKAN LAYANG BERDASARKAN CITRA DAGING MENGGUNAKAN METODE GLCM DAN KNN”, untuk memenuhi salah satu syarat penyusunan Skripsi guna memperoleh gelar sarjana.

Penulis menyadari bahwa dalam penyusunan usulan penelitian ini tidak mungkin terwujud tanpa dukungan, bimbingan dan bantuan dari berbagai pihak, baik bantuan moril maupun materil. Oleh karena itu, dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan banyak terima kasih kepada:

1. Ibu Dr. Hj. Juriko Abdussamad, M.Si, selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo
2. Bapak Dr. Abdul Gafar Latjokke, M.Si, selaku Rektor Universitas Ichsan Gorontalo
3. Bapak Irvan Abraham Salihi, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo
4. Bapak Sudirman Melangi, S.Kom, M.Kom, selaku Wakil Dekan I Bidang Kemahasiswaan 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, S.Kom, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo
7. Bapak Sunarto Taliki, M.Kom, selaku Pembimbing I, yang telah membimbing penulis dalam penyusunan metode penelitian ini

8. Ibu Kartika Chandra Pelangi, M.Kom, selaku Pembimbing II yang telah membantu dan membimbing penulis dalam penyusunan metode penelitian ini
9. Kedua Orang Tua Saya tercinta yang telah memberikan do'a, kasih sayang, dukungan dan semangat yang tak terhingga serta memfasilitasi kebutuhan untuk menyelesaikan skripsi ini
10. Keluarga Besar yang telah memberikan do'a dan support yang tiada henti-hentinya baik dalam segi materil maupun moril
11. Rekan - rekan seperjuangan yang selalu memberikan motivasi kepada penulis
12. Kepada semua pihak yang telah ikut membantu dalam penyusunan penelitian ini yang 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. Akhir kata, semoga usulan penelitian ini dapat bermanfaat bagi kita semua, Aamiin.

Gorontalo, Mei 2023

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
PENGESAHAN SKRIPSI.....	Error! Bookmark not defined.
PERSETUJUAN SKRIPSI	Error! Bookmark not defined.
PERNYATAAN SKRIPSI	Error! Bookmark not defined.
ABSTRACT	v
ABSTRAK	vi
KATA PENGANTAR.....	vii
DAFTAR ISI	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	xiii
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.....	4
1.5.1. Manfaat Teoritis	4
1.5.2. Manfaat Praktis	4
BAB II LANDASAN TEORI.....	5
2.1. Tinjauan Studi.....	5
2.2. Tinjauan Pustaka	6
2.2.1 Ikan Layang	6
2.2.2 Pengolahan Citra Digital	7
2.2.3 Jenis Citra Digital	7
2.2.4 Proses Grayscale.....	8
2.2.5 GLCM (Gray Level Co-Occurrence Matrix)	9
2.2.6 KNN (<i>K-Nearest Neighbor</i>).....	12
2.2.7 Implementasi GLCM dan KNN	13

2.3 Kerangka Pikir	29
BAB III METODE PENELITIAN	30
3.1 Jenis Metode, Subjek, Objek, Waktu dan Lokasi Penelitian	30
3.2 Pengumpulan Data	31
3.3 Pemodelan.....	31
3.3.1 Training Menggunakan KNN	32
3.3.2 Data Testing	32
3.3.3 Model	33
3.3.4 Hasil Klasifikasi	33
3.3.5 Evaluasi	33
BAB IV HASIL PENELITIAN	34
4. 1 Hasil Pengumpulan Data.....	34
4. 2 Hasil Pemodelan.....	34
4.2.1 Pra – Pengolahan	34
4.2.2 Ekstraksi Fitur	36
4.2.3 Klasifikasi	39
BAB V PEMBAHASAN	44
5.1 Pembahasan Model	44
5.2 Pembahasan System	46
BAB VI PENUTUP	50
6.1 Kesimpulan	50
6.2 Saran.....	50
DAFTAR PUSTAKA	ix
Lampiran	51

DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait	5
Tabel 2. 2 Framework rentang 0-15 [24]	17
Tabel 2. 3 Co-Occurrence Matrix [24]	18
Tabel 2. 4 Transpose Matriks.....	19
Tabel 2. 5 Simetris Matriks [24].....	19
Tabel 2. 6 Normalisasi Matriks [24].....	20
Tabel 2. 7 Matriks Kontras [24]	21
Tabel 2. 8 Matriks Homogen [24]	22
Tabel 2. 9 Matriks Entropy [24]	23
Tabel 2. 10 Matriks Energy [24].....	24
Tabel 2. 11 Nilai Hasi Fitur Ekstraksi 0^0 [24].....	24
Tabel 2. 12 Data Fitur Ekstraksi Ciri [24].....	25
Tabel 2. 13 Data Fitur Ekstraksi Baru [24].....	25
Tabel 2. 14 Hasil Pengurutan Fitur [24].....	27
Tabel 2. 15 Kelas Fitur Terdekat [24]	27
Tabel 2. 16 Hasil Akhir [24]	28
Tabel 4. 1 Dataset Ikan Layang	34
Tabel 4. 2 Nilai Dari Suatu Citra	36
Tabel 4. 3 Matrix GLCM	36
Tabel 4. 4 Matrix Ternormalisasi	37
Tabel 4. 5 Hasil Training Data Set	39
Tabel 4. 6 Perhitungan Jarak Euclidean Distance	40
Tabel 4. 7 Hasil Data Testing	41
Tabel 4. 8 Hasil Data Uji K=5	42
Tabel 5. 1 Hasil Evaluasi Nilai	45

DAFTAR GAMBAR

Gambar 2. 1	Tingkat Grayscale [18]	8
Gambar 2. 2	Grayscale [19]	9
Gambar 2. 3	Piksel dengan berbagai sudut [21]	9
Gambar 2. 4	Contoh penentuan awal matriks GLCM [22]	10
Gambar 2. 5	Matriks framework menjadi matriks simetris [22]	10
Gambar 2. 6	Normalisasi matriks GLCM [22]	11
Gambar 2. 7	Penerapan KNN [23]	13
Gambar 2. 8	Citra beras asli [24]	14
Gambar 2. 9	Citra Asli (Kiri) dan Citra Resize (Kanan) [24]	14
Gambar 2. 10	Citra beras melalui Resize [24]	15
Gambar 2. 11	Nilai RGB beras 3x3 [24]	15
Gambar 2. 12	Matrix Greyscale Citra Beras 3x3 [24]	16
Gambar 2. 13	Matriks Greyscale citra Beras range 0-15 [24]	17
Gambar 2. 14	Citra Greyscale range 0-15 [24]	17
Gambar 3. 1	Desain Penelitian	30
Gambar 3. 2	Pemodelan	31
Gambar 5.1	Penghubung Google Drive	46
Gambar 5.2	Kode Akses Penghubung	46
Gambar 5.3	Folder Penghubung	46
Gambar 5.4	File Penghubung	47
Gambar 5.5	Source Code Program Deteksi	47
Gambar 5.6	Script Confusion Matrix	48
Gambar 5.7	Output Confusion Matrix	48
Gambar 5.8	Accuracy Training dan Testing	49

DAFTAR LAMPIRAN

Lampiran 1. 1 Kode Program	51
Lampiran 1. 2 Surat Rekomendasi Penelitian	54
Lampiran 1. 3 Hasil Turnitin	55
Lampiran 1.4 Riwayat Hidup.....	56

BAB I

PENDAHULUAN

1.1. Latar Belakang

Ikan merupakan makanan yang tinggi protein sehingga banyak masyarakat Indonesia termasuk daerah Gorontalo mengkonsumsi ikan sebagai asupan protein untuk kesehatan[1]. Terdapat berbagai jenis ikan dengan nilai ekonomi yang cukup tinggi dalam wilayah perairan Indonesia, diantaranya yaitu cakalang, layang, tuna serta beberapa jenis ikan laut lainnya[2]. Salah satu ikan dengan komoditas hasil perikanan yang cukup banyak adalah ikan layang.

Ikan layang adalah jenis ikan laut yang sering dikonsumsi oleh masyarakat Gorontalo karena harganya yang terjangkau dan memiliki kandungan protein tinggi serta mengandung gizi dan protein yang sangat baik bagi tubuh manusia. Ikan ini juga merupakan komoditi terbesar yang ada di Provinsi Gorontalo[3].

Dalam menjaga kualitas produksi ikan maka diperlukan penanganan yang cepat dan cermat dalam upaya mempertahankan mutunya sejak ikan tersebut diangkat dari air hingga sampai ke tangan konsumen. Permasalahan yang ditemukan adalah ikan layang termasuk komoditas pangan yang mudah rusak dan berdampak pada kualitas kesegaran dagingnya. Ikan yang segar memiliki daging yang keras, tubuh yang kenyal, sisk melekat erat, insang berwarna merah cerah dan elastis. Dibagian luar ikan hanya sedikit atau bahkan tidak berlendir dan berbau segar dan akan tenggelam jika dimasukkan kedalam air[2].

Kesegaran ikan layang selama ini di tentukan dengan cara analisis mikro biologi dan kimiawi. Cara ini di anggap sulit karena membutuhkan tenaga manusia dan cukup melelahkan dan juga pula rentan terhadap kesalahan, mengeluarkan biaya yang besar dan memakan waktu yang cukup lama sehingga dapat memengaruhi produksi kualitas ikan tersebut.

Kemajuan teknologi komputer termasuk interaksi antara manusia dengan komputer pada saat ini sudah berkembang pesat. Bukan hanya di bidang pertanian, bidang perikanan juga sangat membutuhkan kemajuan teknologi guna untuk membantu kelancaran dan meningkatkan hasil produksi. Maka dari itu di butuhkan sebuah aplikasi yang berbasis teknologi pengolahan citra yang memungkinkan untuk memilah dan mengamati proses deteksi kesegaran ikan secara otomatis[5].

Salah satu ciri yang menandakan bahwa ikan tersebut tergolong dalam tingkat kesegaran yang tinggi atau rendah adalah dengan memperhatikan kualitas ciri tekstur daging ikan tersebut. Dalam pengolahan citra, tekstur merupakan salah satu fitur yang dapat di ekstraksi untuk identifikasi citra. Analisis tekstur tidak lepas dari metode klasifikasi. Pengklasifikasian dilakukan untuk mengelompokkan citra berdasarkan pola maupun teksur. Ada beberapa metode pengklasifikasian citra yang dapat digunakan diantaranya adalah metode *K-Nearest Neighbor* (KNN). Metode *K-Nearest Neighbor* (KNN) dilakukan dengan cara klasifikasi citra berdasarkan ekstraksi ciri teksur dengan metode *Gray Level Co-Occurrence Matrix* (GLCM)[7].

GLCM adalah suatu metode statistik yang dapat digunakan untuk analisis tekstur. Matrix Co-Occurence dibentuk dari suatu citra dengan melihat pixel-pixel yang berpasangan yang memiliki intensitas tertentu sehingga dengan menggunakan GLCM, tekstur dengan tingkat kesegaran yang tinggi ataupun rendah dapat dikenali. Metode GLCM menggunakan beberapa fitur pendekatan statistik seperti energi, entropi, kontras, dan sebagainya. Kelebihan metode GLCM yaitu telah terbukti menjadi deskriptor yang efektif serta memiliki akurasi dan waktu komputasi yang lebih baik dari metode ekstraksi lainnya[6].

Berdasarkan beberapa penelitian terkait yang pernah dilakukan oleh Feri Agustina dan Zulfikar Amri Ardiansyah tentang Identifikasi Citra Daging Ayam Kampung dan Broiler Menggunakan Metode GLCM dan Klasifikasi K-NN memperoleh tingkat akurasi sebesar 85%[7]. Dan penelitian dari Irfan Purwanto, M. Afriansyah dan Kusri tentang Deteksi Tingkat Kesegaran Daging Ayam Menggunakan *K-Nearest Neighbor* memperoleh tingkat akurasi sebesar 87%[8].

Adapun penelitian Ayu Kalista, Amin Redjo dan Umi Rosidah tentang Penerapan Image Processing Untuk Tingkat Kesegaran Ikan Nila (*Oreochromis niloticus*) memperoleh kesegaran dengan tingkat akurasi 67,10%[9].

Adapun peneliti ingin mengajukan penelitian tentang “**Deteeksi Kesegaran Ikan Layang Berdasarkan Citra Daging Menggunakan Metode *Gray Level Co-Occurrence Matrix (GLCM)* dan *K-Nearest Neighbor (KNN)***”. Diharapkan penelitian ini dapat memberikan kontribusi, berupa model GLCM dan KNN yang paling akurat untuk mendeteksi tingkat kesegaran ikan layang.

1.2. Identifikasi Masalah

Berdasarkan penjelasan diatas, maka identifikasi masalah dalam penelitian ini yaitu :

1. Terbatasnya fungsi identifikasi secara manual sehingga memberikan hasil yang tidak konsisten untuk mendeteksi tingkat kesegaran ikan layang.
2. Perlunya metode komputerisasi untuk menentukan tingkat kesegaran ikan layang secara otomatis.

1.3. Rumusan Masalah

Berdasarkan identifikasi masalah yang telah di uraikan maka dapat di tentukan rumusan masalah dalam penelitian ini, yaitu:

1. Bagaimana hasil dari metode GLCM dan algoritma KNN untuk bisa mengklasifikasikan kualitas pada ikan layang?
2. Bagaimana kinerja dan efektifitas untuk dapat mengklasifikasikan kualitas ikan layang dengan fitur metode GLCM dan algoritma KNN?

1.4. Tujuan Penelitian

Adapun yang menjadi tujuan pada penelitian ini adalah :

1. Untuk mengetahui hasil akurasi yang di hasilkan dalam mengklasifikasikan kualitas pada ikan layang.

2. Dapat menerapkan teknik pengolahan citra dalam dunia perikanan maupun dunia pemasaran, sehingga berguna bagi masyarakat untuk memperoleh tingkat kesegaran ikan yang relatif lebih bagus.

1.5. Manfaat Penelitian

1.5.1. Manfaat Teoritis

Penelitian ini diharapkan dapat menjadi referensi pada penelitian yang berhubungan dengan deteksi kesegaran ikan layang dengan *image processing* menggunakan metode GLCM dan KNN.

1.5.2. Manfaat Praktis

Sebagai bahan kajian yang melibatkan suatu rancangan *image processing* untuk mendeteksi tingkat kesegaran ikan layang.

BAB II LANDASAN TEORI

2.1. Tinjauan Studi

Penelitian tentang mendeteksi kesegaran ikan dan metode yang digunakan telah dilakukan oleh beberapa peneliti sebelumnya. Berikut beberapa penelitian terkait.

Tabel 2. 1 Penelitian Terkait

N O	PENELITI	JUDUL/TAHUN	METODE	HASIL
1	Suliat Aini, I Gede Pasek Suta Wijaya, Giri Wahyu Wiriasto[10]	Klasifikasi Kesegaran Buah Apel Berdasarkan Ekstraksi Fitur Warna Dan Tekstur Menggunakan Gray Level Co- Occurrence Matrix Dan Algoritma K Nearest Neighbor/2019	Algoritma K-Nearest Neighbor	Pada proses klasifikasi juga dapat disimpulkan, dalam 10 hari pengambilan citra latih dapat dibagi menjadi 3 kelas, yaitu kelas segar untuk hari ke-1 hingga ke-3, kelas kurang segar untuk hari ke-4 hingga ke-6 dan kelas tidak segar untuk hari ke-7 hingga ke-10

2	Muhammad Sarimin, Marta leli Bettiza, Nurul Hayaty, Sapta Nugraha[11]	Implementasi HSV dan GLCM untuk Deteksi Kesegaran Ikan Bawal menggunakan Radial Basis Function Berbasis Android/2019	metode K-Means	Berdasarkan penelitian yang telah dilakukan menggunakan metode K-Means untuk penentuan nilai centroid dapat dilihat bahwa metode k-means memberikan pengaruh terhadap hasil pengenalan yang dilakukan oleh sistem sehingga hasil yang didapatkan lebih baik. Jumlah terbaik yang diperoleh dari perngujian centroid adalah 7 dengan MAD sebesar 0,98.
3	Fittria Shofrotun Ni'mah, T. Sutojo, De Rosal Ignatius Moses Setiadi	Identifikasi Tumbuhan Obat Herbal Berdasarkan Citra Daun Menggunakan Algoritma Gray Level Co-Occurrence Matrix dan K-Nearest Neighbor/2018	Fitur Ekstraksi GLCM dan Klasifikasi KNN	Jumlah terbaik yang diperoleh adalah 83,33%

2.2. Tinjauan Pustaka

2.2.1 Ikan Layang

Ikan layang (*Decapterus russelli*) merupakan salah satu sumberdaya perikanan pelagis ekonomis penting dan sebagai sumber gizi dalam pemenuhan protein hewani, Ikan layang (*Decapterus spp.*) termasuk komponen perikanan pelagis yang penting di Indonesia dan biasanya hidup bergerombol dengan ikan lain seperti lemuru (*Sardinella sirm*), tembang (*Sardinella fimbriala*, *S. perforala*), kembung (*Rastrelliger kanagaa*. R.

brachysoma), selar (Canax sp.), dan ekor kuning (Caerio sp.). (Prihartini, 2006).

2.2.2 Pengolahan Citra Digital

Pengolahan citra digital adalah pemrosesan citra, khususnya dengan menggunakan komputer sehingga citra itu kualitasnya menjadi lebih baik dan menghasilkan informasi nilai untuk tiap-tiap warnanya. Pengolahan citra dapat didefinisikan dengan pengertian computer vision yang dikaitkan dengan perolehan citra, pemrosesan, klasifikasi, penganan, dan pencakupan keseluruhan, pengambilan keputusan yang diikuti pengidentifikasian citra [13]. Pengolahan citra digital adalah proses yang bertujuan untuk memanipulasi dan menganalisis citra menggunakan bantuan komputer. Pengolahan citra digital dapat dikelompokkan dalam dua aspek kegiatan :

1. Memperbaiki kualitas gambar, agar dapat lebih mudah diinterpretasi oleh mata manusia.
2. Mengolah informasi yang terdapat pada suatu gambar untuk keperluan pengenalan objek secara otomatis.

Aspek kedua sangat erat hubungannya dengan ilmu pengenalan pola (pattern recognition) yang umumnya bertujuan mengenali suatu objek dengan cara mengekstrak informasi penting yang terdapat pada suatu citra [14].

2.2.3 Jenis Citra Digital

Pada aplikasi pengolahan citra digital umumnya, citra digital dibagi menjadi 3, RGB image, Grayscale image dan Binary image [15] :

a. RGB

Citra berwarna atau citra RGB adalah citra yang menyediakan beberapa warna dalam bentuk komponen R (merah), G (hijau), dan B (biru). Masingmasing komponen menggunakan delapan bit yang nilainya berkisar antara 0 sampai dengan 255 [16].

b. *Grayscale*

Grayscale adalah suatu citra citra yang hanya memiliki warna tingkat keabuan. Gambar grayscale 8 bit memiliki 256 tingkat warna abu-abu mulai dari putih hingga hitam [17]. Tingkat Grayscale dapat di lihat pada gambar 2.1



Gambar 2. 1 Tingkat Grayscale [18]

Untuk mendapatkan nilai grayscale dapat dilakukan dengan mengambil rata-rata dari nilai R, G dan B [18]. seperti di bawah ini :

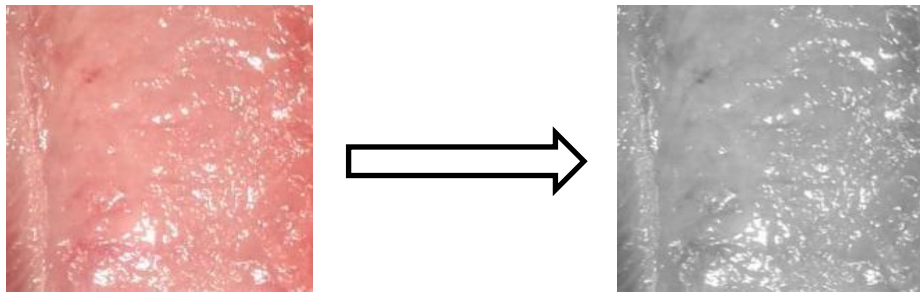
$$s = \frac{r + g + b}{3} \quad (1)$$

c. *Binary Image*

Binary image terdiri dari warna hitam atau putih, karena hanya ada dua warna untuk per piksel, maka hanya perlu 1 bit setiap piksel (0 dan 1) atau apabila dalam 8 bit (0 dan 255), sehingga sangat efisien dalam hal penyimpanan. atau gambar arsitektur. *Binary image* merupakan hasil pengolahan dari *black and white image* [15].

2.2.4 Proses Grayscale

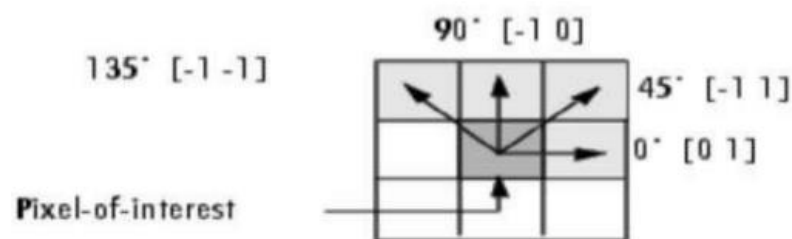
Proses grayscale adalah proses dimana citra input yang terdiri dari tiga layer warna RGB dirubah hanya menjadi satu layer, dimana layer tersebut menyatakan tingkat keabuan disetiap titik pada citra. Proses ini diperlukan karena pada proses ekstraksi ciri GLCM memerlukan citra grayscale sebagai input-nya. Proses perubahan ini dilakukan dengan melakukan penjumlahan komponen RGB yang telah diberikan bobot masing-masing [19].



Gambar 2. 2 Grayscale [19]

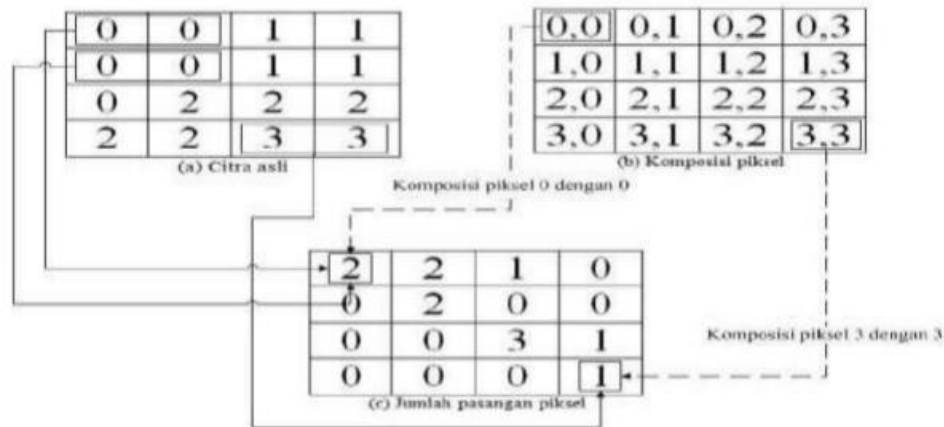
2.2.5 GLCM (Gray Level Co-Occurrence Matrix)

Gray Level Co-occurrence Matrix (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 [20]. Gray Level Co-Occurance Matrices (GLCM) adalah salah satu metode analisis tekstur orde kedua. GLCM pertama kali diusulkan oleh Haralick pada tahun 1973. GLCM merupakan suatu metode untuk melakukan ekstraksi ciri berbasis statistik, perolehan ciri diperoleh dari nilai piksel matrix yang mempunyai nilai tertentu dan membentuk suatu sudut pola [12]. GLCM merepresentasikan hubungan dua piksel yang bertetangga dimana dua piksel yang berhubungan tersebut memiliki intensitas keabuan tertentu serta memiliki jarak dan arah tertentu diantara keduanya. Jarak dinyatakan piksel dan arah dinyatakan sudut. Jarak dapat bernilai 1, 2, 3 dan seterusnya sedangkan arah dapat bernilai 0° , 45° , 90° , 135° dan seterusnya [12].



Gambar 2. 3 Piksel dengan berbagai sudut [21].

Sebagai ilustrasi, ketetanggaan piksel yang dapat dipilih ke arah timur (kanan). Salah satu untuk merepresentasikan hubungan ini yaitu berupa (1,0), yang menyatakan hubungan dua piksel bernilai 1 diikuti dengan piksel bernilai 0, sehingga jumlah kelompok piksel memenuhi hubungan tersebut dihitung [12].



Gambar 2. 4 Contoh penentuan awal matriks GLCM [22]

Matriks pada Gambar 2.4 dinamakan matrix framework. Matriks ini kemudian diolah menjadi matriks dimetris dengan cara menambahkan hasil transposnya [12]. Seperti yang terlihat pada Gambar 2.5

$$\begin{bmatrix} 2 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

Tranpos
GLCM sebelum dinormalisasi

Gambar 2. 5 Matriks framework menjadi matriks simetris [22].

Untuk menghilangkan ketergantungan pada ukuran citra, nilai-nilai elemen GLCM perlu dinormalisasikan sehingga jumlahnya bernilai 1. Dengan demikian, hasil normalisasi dari matriks GLCM pada Gambar 2.6

$$\begin{bmatrix} \frac{4}{24} & \frac{2}{24} & \frac{1}{24} & \frac{0}{24} \\ \frac{2}{24} & \frac{4}{24} & \frac{0}{24} & \frac{0}{24} \\ \frac{1}{24} & \frac{0}{24} & \frac{6}{24} & \frac{1}{24} \\ \frac{0}{24} & \frac{0}{24} & \frac{1}{24} & \frac{2}{24} \end{bmatrix}$$

Gambar 2. 6 Normalisasi matriks GLCM [22]

Untuk mendapatkan fitur tekstur GLCM, hanya 14 besaran yang diusulkan oleh Haralick (1973) untuk dipakai. Beberapa fitur yang akan dipakai nantinya adalah *contrast*, *homogeneity*, *entropy*, *energy* sebagai berikut :

1. *Contrast* digunakan untuk mengukur frekuensi spasial dari citra dan perbedaan momen GLCM. Perbedaan yang dimaksudkan adalah perbedaan tinggi dan rendah pixel, Contrast merupakan ukuran keberadaan variasi atas keabuan pixel citra dihitung.

$$Contrast = \sum_{i1} \sum_{i2} (i1 - i2)^2 p(i_1, i_2) \quad (2)$$

2. *Homogeneity* digunakan untuk mengukur homogenitas yaitu ukuran kedekatan distribusi masing-masing elemen pada matriks GLCM ke matriks GLCM diagonal. Homogeneity dihitung dengan cara berikut :

$$Homogeneity = \sum_{i1} \sum_{i2} \frac{p(i_1, i_2)}{1 + |i_1 - i_2|} \quad (3)$$

3. Entropy adalah mengukur heterogenitas dan berkorelasi dengan standar deviasi. Penghitungan Entrophy digunakan untuk mengukur kompleksitas (keacakan) citra, entropy akan bernilai tinggi ketika citra tidak seragam. Entropy dihitung dengan cara berikut :

$$Entropy = - \sum_{i_1} \sum_{i_2} p(i_1, i_2) \log p(i_1, i_2) \quad (4)$$

4. Energy adalah pengukuran intensitas keseragaman piksel. Sebuah keadaan homogen mengandung hanya sedikit gray level tetapi memiliki nilai piksel Pij yang tinggi, oleh karena itu jumlah dari pangkat Pij akan tinggi.

$$Energy = \sum_{i_1} \sum_{i_2} p^2(i_1, i_2) \quad (5)$$

2.2.6 KNN (*K-Nearest Neighbor*)

Algoritma *K-Nearest Neighbor* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data pembelajaran. Sebuah titik pada ruang ini ditandai kelas c jika kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan jarak Euclidean dengan rumus seperti pada persamaan [16].

$$Distance = \sqrt{\sum_{i=1}^n (X_{training}^i - X_{testing})^2} \quad (6)$$

Yaitu:

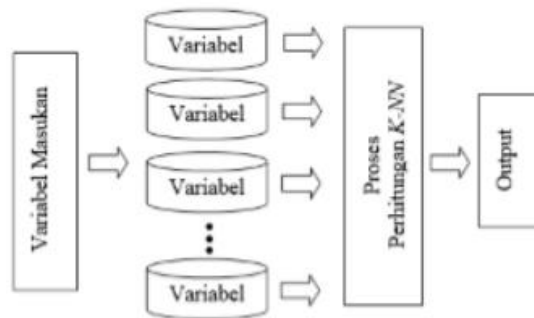
$X_{training}^i$: Data training ke- i

$X_{testing}$: Data Testing

i : Record (baris) ke- i dari table

n : Jumlah data training

Penerapan Metode *K-Nearest Neighbor*. Adapun penerapan metode K-NN melalui beberapa langkah [23]:



Gambar 2. 7 Penerapan KNN [23]

1. Tentukan parameter k
2. Hitung jarak antara data yang akan dievaluasi dengan semua pelatihan
3. Urutkan jarak yang terbentuk (urut naik)
4. Tentukan jarak terdekat sampai urutan k
5. Pasangkan kelas yang bersesuaian
6. Cari jumlah kelas dari tetangga yang terdekat dan tetapkan kelas tersebut sebagai kelas data yang akan dievaluasi.

2.2.7 Implementasi GLCM dan KNN

Contoh studi kasus :

Berdasarkan studi kasus dalam penelitian sebelumnya pernah melakukan penelitian dengan judul “Penerapan Algoritma *Grey Level Co Occurrence Matrix* (GLCM) Dengan Metode *K-Nearest Neighbour* Untuk Klasifikasi Beras Berdasarkan Tekstur” studi kasus kali ini akan melakukan penelitian dengan analisis, dan mengkombinasikan algoritma ekstraksi ciri *grey level co-occurrence matrix* dengan metode klasifikasi *k-nearest neighbor* untuk mengklasifikasikan citra dan menguji tingkat keakuratan dalam pengujian [24].

2.2.7.1 Analisis Perhitungan Data Masukan

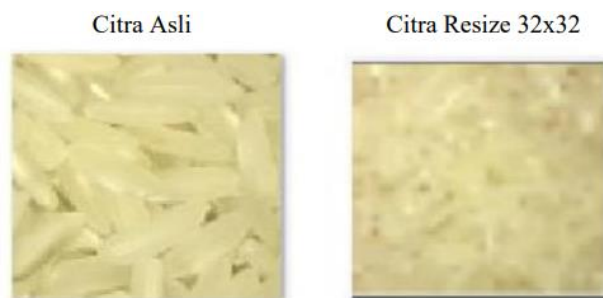
Data masukan berupa citra beras dengan berbagai kualitas dan akan melalui beberapa tahapan proses pengolahan citra, diantaranya adalah proses *resizing*, *grayscale*, kuantisasi citra grayscale dan ekstraksi ciri. Berikut data masukan :



Gambar 2. 8 Citra beras asli [24]

1. Resizing

Untuk proses yang lebih cepat dalam mencari parameter pada citra data masukan, citra masukan awal akan di ubah ukurannya dengan proses resizing dari ukuran citra asli menjadi citra ukuran 32 x 32 pixel dengan mode warna RGB (Red, Green, Blue) yang artinya dalam setiap pixel terdapat 3 nilai warna. Berikut adalah gambar citra asli(kiri) dan citra hasil resize(kanan) :



Gambar 2. 9 Citra Asli (Kiri) dan Citra Resize (Kanan) [24]

2. Grayscale

Setelah melalui tahap resize, maka tahap selanjutnya adalah mengubah citramode RGB, menjadi Greyscale. Proses ini bertujuan untuk menyederhanakan nilai pixel pada sebuah citra. Yang awalnya pada setiap pixel memiliki 3 nilai yaitu RGB menjadi hanya 1 nilai keabuan. Rumus yang digunakan untuk proses greyscale adalah sebagai berikut :

$$X = 0.21 * R + 0.72 * G + 0.07 * B$$

(7)

Dimana

X = Nilai Greyscale

R = Nilai Red

G = Nilai Green

B = Nilai Blue

Analisis perhitungan :

Agar dapat melakukan tahap perhitungan grayscale maka dibutuhkan citra masukan, berikut ini merupakan citra masukan data 1, yaitu citra beras yang telah dilakukan proses resizing:



Gambar 2. 10 Citra beras melalui Resize [24]

A=,255, R=,220, G=,204, B=,143	A=,255, R=,219, G=,205, B=,144	A=,255, R=,208, G=,189, B=,130
A=,255, R=,214, G=,198, B=,136	A=,255, R=,216, G=,202, B=,141	A=,255, R=,240, G=,226, B=,165
A=,255, R=,224, G=,208, B=,146	A=,255, R=,209, G=,193, B=,133	A=,255, R=,253, G=,243, B=,181

Gambar 2. 11 Nilai RGB beras 3x3 [24]

Perhitungan nilai Matriks :

Perhitungan piksel (0.0) :

$$X = 0,21 * 220 + 0,72 * 204 + 0,07 * 143$$

$$X = 46,2 + 146,88 + 10,1$$

$$X = 203,18 = 203$$

Perhitungan piksel (0.1) :

$$X = 0,21 * 219 + 0,72 * 205 + 0,07 * 144$$

$$X = 45,99 + 147,6 + 10,8$$

$$X = 203,67 = 204$$

Perhitungan piksel (0.2) :

$$X = 0,21 * 208 + 0,72 * 189 + 0,07 * 130$$

$$X = 43,68 + 136,8 + 9,1$$

$$X = 188,86 = 189$$

Dengan menggunakan rumus yang sama pada semua piksel maka akan di dapatkan hasil matriks sebagai berikut :

203	204	189
197	201	220
207	192	241

Gambar 2. 12 Matrix Greyscale Citra Beras 3x3 [24]

3. Kuantisasi

Setelah dilakukan konversi dari mode RGB ke greyscale, maka akan didapatkan 1 nilai derajat keabuan dengan rentang nilai keabuan 0-255, selanjutnya akan dilakukan proses kuantisasi citra yang sudah melalui tahap greyscale dengan rentang nilai 0-255 menjadi 0-15. Proses ini bertujuan agar menghemat waktupemrosesan komputasi. Rumus yang digunakan untuk proses penskalaan citra greyscale nilai keabuan 0-255 menjadi 0-15 adalah dengan persamaan berikut :

$$X_b = \text{int}(X * (2^n - 1) / 255)$$

Dimana :

X_b = Nilai Greyscale Baru

X = Nilai Greyscale Lama

2^n = Nilai Derajat Keabuan Baru

$$X_b = \text{int}(203 * 15 / 255) = 11,94$$

$$X_b = 12$$

Perhitungan piksel (0,1) :

$$X_b = \text{int}(204 * 15 / 255) = 12$$

$$X_b = 12$$

Perhitungan piksel (0,1) :

$$X_b = \text{int}(189 * 15 / 255) = 12,29$$

$$X_b = 11$$

M(G)	0	1	2
0	12	12	11
1	12	12	4
2	13	11	14

Gambar 2. 13 Matriks Greyscale citra Beras range 0-15 [24]

2.2.7.2 Ekstraksi Co-occurrence Matrix

Metode Matrix Co-occurrence merupakan metode untuk ekstraksi ciri, dimana nilai ekstraksi ciri yang akan dicari adalah nilai Contrasts, Homogenitas, energy, dan dissimilarity. Sebagai contoh, citra kacang hijau dibawah ini telah melalui 38 preprocessing dengan ukuran 3x3 pixel yang memiliki 16 derajat keabuan dengan rentang dari 0-15.

Greyscale 0-15



Gambar 2. 14 Citra Greyscale range 0-15 [24]

Langkah-langkah untuk ekstraksi ciri dengan metode co-occurrence matrix adalah:

1. Membuat area kerja atau framework

memiliki 16 derajat keabuan, maka jumlah piksel ketetanggaan dan nilai pixel referensi pada area kerja matriks berjumlah 16, seperti pada tabel 2.2 berikut :

Tabel 2. 2 Framework rentang 0-15 [24]

P	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	0,10	0,11	0,12	0,13	0,14	0,15
1	1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	1,10	1,11	1,12	1,13	1,14	1,15
2	2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9	2,10	2,11	2,12	2,13	2,14	2,15
3	3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9	3,10	3,11	3,12	3,13	3,14	3,15
4	4,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9	4,10	4,11	4,12	4,13	4,14	4,15
5	5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9	5,10	5,11	5,12	5,13	5,14	5,15
6	6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9	6,10	6,11	6,12	6,13	6,14	6,15
7	7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9	7,10	7,11	7,12	7,13	7,14	7,15
8	8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9	8,10	8,11	8,12	8,13	8,14	8,15
9	9,0	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9	9,10	9,11	9,12	9,13	9,14	9,15
10	10,	10,1	10,2	10,3	10,4	10,5	10,6	10,7	10,8	10,9	10,10	10,11	10,12	10,13	10,14	10,15
11	11,0	11,1	11,2	11,3	11,4	11,5	11,6	11,7	11,8	11,9	11,10	11,11	11,12	11,13	11,14	11,15
12	12,0	12,1	12,2	12,3	12,4	12,5	12,6	12,7	12,8	12,9	12,10	12,11	12,12	12,13	12,14	12,15
13	13,0	13,1	13,2	13,3	13,4	13,5	13,6	13,7	13,8	13,9	13,10	13,11	13,12	13,13	13,14	13,15
14	14,0	14,1	14,2	14,3	14,4	14,5	14,6	14,7	14,8	14,9	14,10	14,11	14,12	14,13	14,14	14,15
15	15,0	15,1	15,2	15,3	15,4	15,5	15,6	15,7	15,8	15,9	15,10	15,11	15,12	15,13	15,14	15,15

2. Membentuk Co-Occurrence Matrix

untuk proses selanjutnya adalah menghitung nilai dari matriks dengan mengisi jumlah hubungan spasial. Matriks kookurensi yang akan dibuat adalah hubungan spasial untuk $d=1$ dan sudut 0^0 . Matriks dibuat dengan mengisi jumlah hubungan sasial yang ada matriks grayscale. Maka didapatkanlah matriks kookurensi seperti pada tabel berikut :

Tabel 2. 3 Co-Occurrence Matrix [24]

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	11	5	3	1	0
11	0	0	0	0	0	0	0	0	1	0	9	96	77	16	0	0
12	0	0	0	0	0	0	0	0	1	2	12	80	278	75	7	0
13	0	0	0	0	0	0	0	0	0	0	2	12	83	192	8	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1. Mentranspose Co-Occurrence Matrix

Yang sudah didapat selanjutnya di transpose, transpose merupakan proses mengubah matriks baris menjadi kolom, tujuan dari proses ini adalah untuk mendapatkan sudut simetrisnya yaitu sudut 180^0 . Berikut merupakan hasil pembentukan matriks transpose :

Tabel 2. 4 Transpose Matriks

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	2	0	1	1	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	9	12	2	0	0
11	0	0	0	0	0	0	0	0	0	0	11	96	80	12	1	0
12	0	0	0	0	0	0	0	0	0	2	5	77	278	83	7	0
13	0	0	0	0	0	0	0	0	0	0	3	16	75	192	8	0
14	0	0	0	0	0	0	0	0	0	0	1	0	7	8	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2. Membuat Matriks Simetris

Pada proses ini adalah menjumlahkan co-occurrence matriks dengan transpose matriks agar membentuk simetris antara sudut 0^0 dan sudut 180^0 . Seperti matriks berikut ini :

Tabel 2. 5 Simetris Matriks [24]

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	4	0	1	3	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0	4	0	0	0
10	0	0	0	0	0	0	0	0	0	0	2	20	17	5	1	0
11	0	0	0	0	0	0	0	0	1	0	20	192	157	28	1	0
12	0	0	0	0	0	0	0	0	0	4	17	157	556	158	14	0
13	0	0	0	0	0	0	0	0	0	0	5	28	158	384	16	0
14	0	0	0	0	0	0	0	0	0	0	1	1	14	16	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7. Membuat Matriks Normalisasi

Untuk mendapatkan nilai dari normalisasi yaitu dengan cara menjumlahkan semua matriks simetris, kemudiandijadikan pembagi untuk semua piksel yang ada pada matriks simetris. untuk mencari probabilitas maka digunakan perhitungan $P(i,j) = (i,j) / \text{total jumlah pasangan}$

Dimana total jumlah pasangan = 1986

Contoh perhitungan $P(12,9) = 4/1985 = 0,002014$

Berikut tabel matriks hasil perhitungan normalisasi:

Tabel 2. 6 Normalisasi Matriks [24]

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0,002014	0	0,000504	0,001511	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0,000504	0	0	0,002014	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0,001007	0,001007	0,00856	0,002518	0,00504	0
11	0	0	0	0	0	0	0	0	0,000504	0	0,01007	0,096677	0,079053	0,01499	0,00504	0
12	0	0	0	0	0	0	0	0	0	0,002014	0,00856	0,079053	0,27966	0,079557	0,007049	0
13	0	0	0	0	0	0	0	0	0	0	0,002518	0,014099	0,079557	0,193353	0,008056	0
14	0	0	0	0	0	0	0	0	0	0	0,002014	0,002014	0,07049	0,008056	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2.2.7.3 Ekstraksi Ciri Fitur

Setelah didapatkan nilai co-occurrence matriks maka proses selanjutnya adalah menghitung nilai ciri statistiknya. Ciri statistik yang akan di hitung adalah kontras, homogenitas, energy, entropy dan dissimilarity. Berikut adalah perhitungan fitur tekstur :

1. Perhitungan Kontras

$$Contrast = \sum_{i1} \sum_{i2} (i1 - i2)^2 p(i_1, i_2) \quad (2)$$

Dimana :

i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatan (i,j) adalah Probabilitas kolom (i,j)

Syarat : Ketika nilai i dan j sama, sel berada pada diagonal dan $(i-j) = 0$. Nilai-nilai ini merepresentasikan pixel yang keseluruhannya mirip dengan tetangga mereka, sehingga mereka diberi bobot 0

Contoh perhitungan :

$$P(12,9).(12-9)^2 = 0,002014 \cdot 9 = 0,018126$$

Berikut tabel hasil perhitungan kontras :

Tabel 2. 7 Matriks Kontras [24]

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0,0013609	0,008065	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0,018145	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0,010081	0,034274	0,022681	0,008056	0
11	0	0	0	0	0	0	0	0	0,013609	0	0,010081	0	0,079133	0,056452	0,004536	0
12	0	0	0	0	0	0	0	0	0,008065	0,018145	0,034274	0,079133	0	0,079637	0,028226	0
13	0	0	0	0	0	0	0	0	0	0	0,022681	0,056452	0,079637	0	0,008056	0
14	0	0	0	0	0	0	0	0	0	0	0,008065	0,008065	0,078776	0,008056	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1. Perhitungan Homogenitas

$$Homogeneity = \sum_{i_1} \sum_{i_2} \frac{p(i_1, i_2)}{1 + |i_1 - i_2|} \quad (3)$$

Dimana :

i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatan $p(i,j)$ adalah Probabilitas kolom (i,j)

Contoh perhitungan :

$$P(12,9) / [1+(12-9)^2] = 0,002014/10 = 0,000201$$

Tabel 2. 8 Matriks Homogen [24]

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0,000151	2,96E-05	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0,000202	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0,001008	0,00504	0,001714	0,000252	2,96E-05	0
11	0	0	0	0	0	0	0	0	0,000151	0	0,00504	0,096774	0,039567	0,002823	5,04E-05	0
12	0	0	0	0	0	0	0	0	2,96E-05	0,000202	0,001714	0,039567	0,280242	0,039818	0,001411	0
13	0	0	0	0	0	0	0	0	0	0	0,000252	0,002823	0,039818	0,193548	0,004032	0
14	0	0	0	0	0	0	0	0	0	0	2,96E-05	5,04E-05	0,001411	0,004032	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2. Perhitungan Entropy

$$Entropy = - \sum_{i1} \sum_{i2} p(i_1, i_2) \log p(i_1, i_2) \quad (4)$$

Dimana :

i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatan p (i,j) adalah Cooccurrence Matrix Simetris Ternormalisasi

Contoh perhitungan :

$$\begin{aligned}
 &-(P(i,j) \log (P(i,j))) \\
 &= -(0,002014 \cdot \log(0,002014)) \\
 &= 0.00542962
 \end{aligned}$$

Tabel 2. 9 Matriks Entropy [24]

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0,004265	0,001662	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0,005434	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0,003021	0,020126	0,017712	0,006549	0,001662	0
11	0	0	0	0	0	0	0	0	0,004265	0	0,020126	0,098152	0,087176	0,026114	0,001662	0
12	0	0	0	0	0	0	0	0	0,001662	0,005429	0,017712	0,087176	0,154824	0,087512	0,015181	0
13	0	0	0	0	0	0	0	0	0	0	0,006549	0,026114	0,087512	0,138041	0,016882	0
14	0	0	0	0	0	0	0	0	0	0	0,001662	0,001662	0,015181	0,016882	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3. Perhitungan Energy

$$Energy = \sum_{i1} \sum_{i2} p^2 (i_1, i_2) \quad (5)$$

Dimana :

i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatanp (i,j) adalah Cooccurence Matrix Simetris Ternormalisasi

Contoh perhitungan :

$$P(i,j)^2 = (0,002014)^2 = 4.0562E-06$$

Tabel 2. 10 Matriks Energy [24]

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	2,96E-06	2,96E-07	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	4,06E-06	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1,02E-06	0,000102	7,34E-05	6,35E-06	2,96E-07	0
11	0	0	0	0	0	0	0	0	2,96E-06	0	0,000102	0,009365	0,006262	0,000199	2,96E-07	0
12	0	0	0	0	0	0	0	0	2,96E-07	0,000201	7,34E-05	0,006262	0,078536	0,006342	4,98E-05	0
13	0	0	0	0	0	0	0	0	0	0	6,35E-06	0,000199	0,006342	0,037461	650E-05	0
14	0	0	0	0	0	0	0	0	0	0	2,96E-07	2,96E-07	4,06E-06	650E-05	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Setelah semua proses perhitungan fitur dilakukan, langkah selanjutnya adalah menjumlahkan tiap matriks dari fitur tersebut sehingga didapatkan hasil sebagai berikut [25] :

Tabel 2. 11 Nilai Hasi Fitur Ekstraksi 0⁰[24]

NO	Pixel	Kontras	Homogenitas	Entropy	Energy
1	Sudut 0	0,74193548	0,76181214	0,97791544	0,15157602

Langkah selanjutnya adalah merata-ratakan nilai nilai-nilai dari semua sudut agar didapatkan satu nilai tunggal dari setiap fitur yang telah didapatkan sehingga memudahkan dalam penamaan klasifikasi.

2.2.7.4 Analisis K-Nearest Neighbor

K-Nearest Neighbor merupakan sebuah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumnya. KNN termasuk dalam golongan supervised , dimana hasil query instance yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori yang ada dalam KNN. Nantinya kelas yang baru dari suatu data akan dipilih berdasarkan grup kelas yang dekat jarak vektornya [25].

Berikut ini merupakan data fitur dari ekstraksi ciri yang telah dilakukan :

Tabel 2. 12 Data Fitur Ekstraksi Ciri [24]

No	Kontras	Homogen	Entropy	Energy	Kualitas
1	0,84063	0,73549594	0,99991	0,1402449	KW1
2	0,86502	0,71461713	0,9921	0,157529	KW2
3	0,84176	0,73543265	1,02263	0,1508495	KW2
4	0,71875	0,75252378	0,97816	0,1719445	KW3
5	0,99741	0,72354421	1,06102	0,1468904	KW3
6	0,88041	0,72738139	0,99813	0,1684139	KW3
7	0,78626	0,74458208	1,01596	0,1491679	KW3
8	0,78768	0,73004611	1,05025	0,1250305	KW3
9	0,7671	0,74833839	0,97848	0,1772233	KW3
10	0,82856	0,73098391	1,0254	0,1498693	KW3
11	0,86661	0,73047063	1,0016	0,1583219	KW3
12	1,09934	0,70566359	1,06476	0,1336976	KW3
13	0,7745	0,73378537	1,00315	0,1511677	KW3
14	0,98534	0,72181986	1,06477	0,1383272	KW1
15	0,89584	0,72037709	1,03308	0,1460091	KW1
16	0,88325	0,72378444	1,0434	0,1371562	KW1
17	0,94905	0,71179787	1,10545	0,1143853	KW1
18	0,96721	0,70730541	1,08783	0,1299694	KW1
19	0,93986	0,7051477	1,04588	0,1281893	KW1
20	1,14044	0,70003985	1,04454	0,14616114	KW2
21	0,82856	0,73098391	1,0254	0,1498693	KW2
22	0,7671	0,74833839	0,97848	0,1772233	KW2
23	0,78768	0,73004611	1,05025	0,1250305	KW2
24	0,78626	0,74458208	1,01596	0,1491679	KW2
25	0,88041	0,72738139	0,99813	0,1684139	KW2

Data uji : terdapat nilai fitur citra baru yaitu sebagai berikut :

Tabel 2. 13 Data Fitur Ekstraksi Baru [24]

Kontras	Homogen	Entropy	Energy	Kualitas
0,71875	0,752523781	0,97816	0,17194454	?

Langkah pertama yang dilakukan adalah menentukan K tetangga terdekat. Nilai K yang terbaik untuk metode ini tergantung pada data. Secara umum nilai K yang tinggi akan mengurangi efek noise pada klasifikasi tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur [26]. Untuk itu kita ambil nilai K adalah $\frac{1}{4}$ dari data uji. Dalam contoh kasus ini adalah $6,5 \approx 6$ sehingga, selanjutnya

hitung kuadrat jarak euclid pada masing-masing objek dengan rumus euclidean distance yaitu :

$$Distance = \sqrt{\sum_{i=1}^n (X_{training}^i - X_{testing})^2}$$

Keterangan :

d = jarak data uji ke data pembelajaran.

x_j = data uji ke- j , dengan $j = 1, 2, \dots n$.

y_j = data pembelajaran ke- j dengan $j = 1, 2, \dots n$.

Urutkan berdasarkan nilai fitur terkecil :

Tabel 2. 14 Hasil Pengurutan Fitur [24]

NO	Kontras	Homogen	Entropy	Energy	Kualitas	Euclidean
1	0,84063	0,73549594	0,99991359	0,14024493	KW1	2,60280244
4	0,71875	0,75252378	0,97815542	0,17194454	KW3	2,352008903
9	0,7671	0,74833839	0,97848427	0,17722334	KW3	2,43512448
22	0,7671	0,74833839	0,97848427	0,17722334	KW2	2,43512448
13	0,7745	0,73378537	1,00315364	0,15116771	KW3	2,49533117
7	0,78626	0,74458208	1,01595654	0,14916792	KW3	2,536172605
24	0,78626	0,74458208	1,01595654	0,14916792	KW2	2,536172605
8	0,78768	0,73004611	1,05024585	0,12503055	KW3	2,609082919
23	0,78768	0,73004611	1,05024585	0,12503055	KW2	2,609082919
10	0,82856	0,73098391	1,02539607	0,24986927	KW3	2,6380288
21	0,82856	0,73098391	1,02539607	0,14986927	KW2	2,6380288
2	0,86502	0,71461713	0,99210331	0,15752899	KW2	2,651580422
3	0,84176	0,73543265	1,02263229	0,15084949	KW2	2,655297614
11	0,86661	0,73047063	1,00160353	0,15832186	KW3	2,662319219
6	0,88041	0,72738139	0,99812988	0,1684139	KW3	2,689514886
25	0,88041	0,72738139	0,99812988	0,1684139	KW2	2,689514886
16	0,88325	0,72378444	1,04339988	0,13715622	KW1	2,780189458
15	0,89584	0,72037709	1,03308042	0,14600906	KW1	2,787715194
19	0,93986	0,7051477	1,04587929	0,12818929	KW1	2,908480404
14	0,98534	0,72181986	1,0647745	0,13832716	KW1	3,45681829
17	0,94905	0,71179787	1,1054542	0,114385825	KW1	3,048883632
18	0,96721	0,70730541	1,08782619	0,12996939	KW1	3,056633328
5	0,99741	0,72354421	1,06102021	0,1468904	KW3	3,057084841
12	1,09934	0,70566359	1,06476113	0,13369764	KW3	3,305396907
20	1,14044	0,70003985	1,04453961	0,14616138	KW2	3,375301213

Dengan nilai K yang telah ditentukan sebelumnya, maka kelas-kelas terdekat adalah seperti pada tabel berikut :

Tabel 2. 15 Kelas Fitur Terdekat [24]

NO	Kontras	Homogen	Entropy	Energy	Kualitas	Euclidean
1	0,84063	0,73549594	0,99991359	0,14024493	KW1	2,60280244
4	0,71875	0,75252378	0,97815542	0,17194454	KW3	2,352008903
9	0,7671	0,74833839	0,97848427	0,17722334	KW3	2,43512448
22	0,7671	0,74833839	0,97848427	0,17722334	KW2	2,43512448
13	0,7745	0,73378537	1,00315364	0,15116771	KW3	2,49533117
7	0,78626	0,74458208	1,01595654	0,14916792	KW3	2,536172605

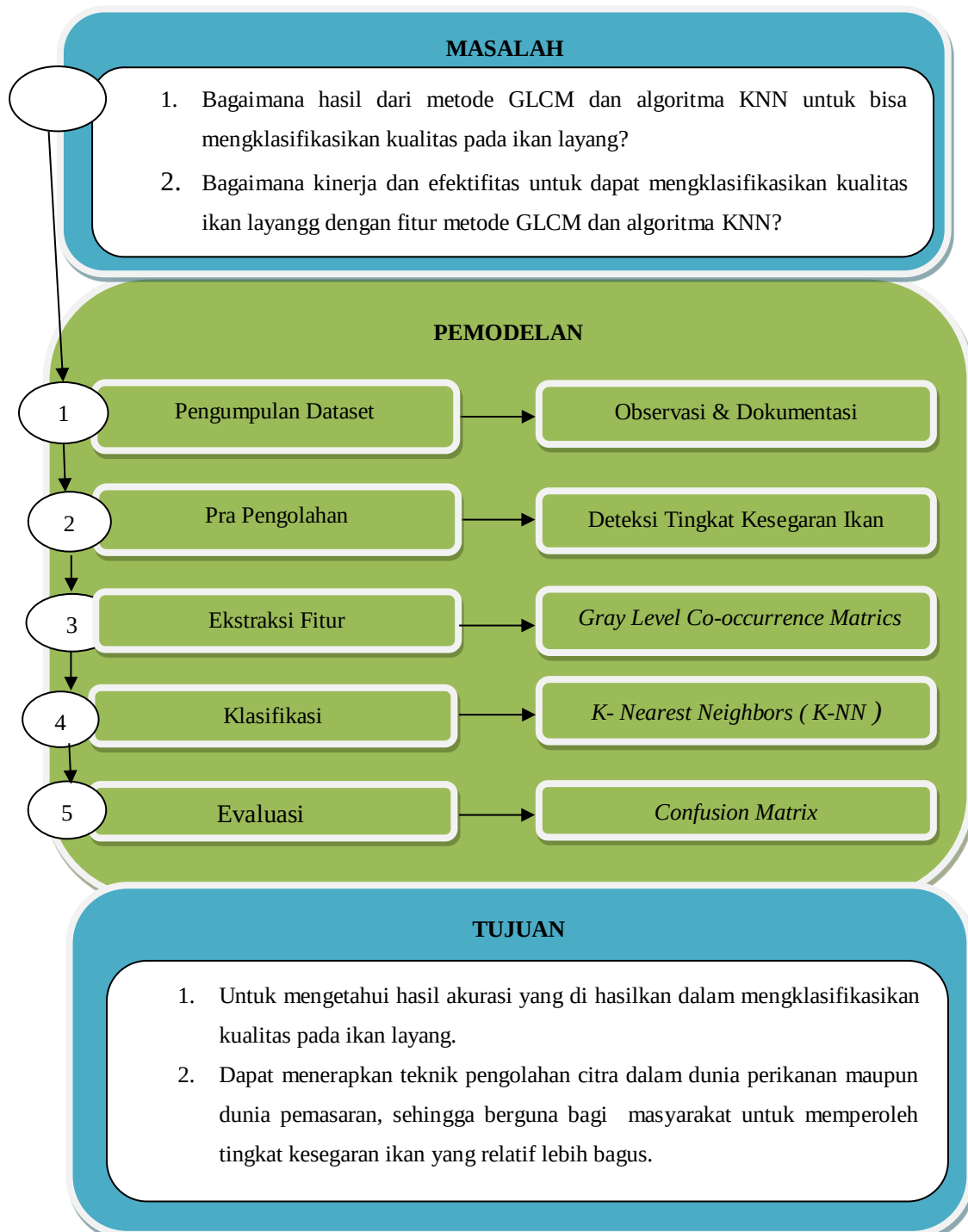
Karena mayoritas kelasnya adalah KW3, maka data baru yang menjadi perbandingan termasuk kelas KW3 [24].

Tabel 2. 16 Hasil Akhir [24]

Kontras	Homogen	Entropy	Energy	Kualitas
0,71875	0,752523781	0,97816	0,17194454	KW3

Jadi kesimpulan dari contoh studi kasus di atas adalah hasil dari kualitas yang di cari adalah KW3 karena perbandingan dengan kelas baru yang jumlahnya hanya sedikit di banding dengan kualitas KW3, jadi hasil dari output pencarian kualitas di atas adalah mayoritas Kualitas KW3.

2.3 Kerangka Pikir



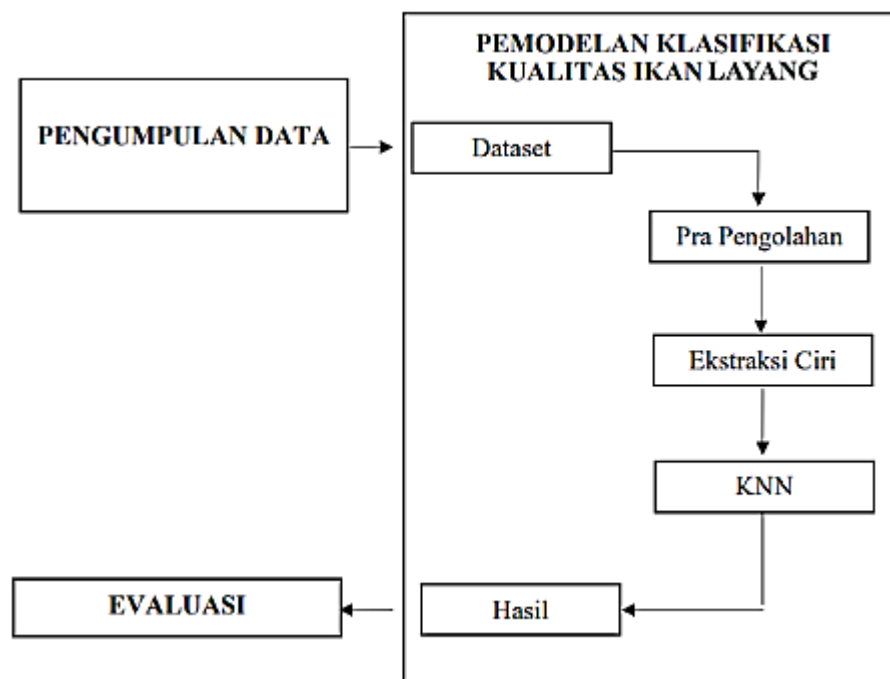
BAB III

METODE PENELITIAN

3.1 Jenis Metode, Subjek, Objek, Waktu dan Lokasi Penelitian

Dilihat dari tingkat penerapannya maka penelitian ini merupakan penelitian terapan. Dipandang dari jenis data yang diperoleh maka penelitian ini merupakan penelitian kuantitatif. Dipandang dari perlakuan terhadap data maka penelitian konfirmatori. Penelitian ini menggunakan metode penelitian eksperimen. Dengan demikian jenis penelitian ini adalah penelitian eksperimental.

Subjek penelitian ini adalah klasifikasi pada objek daging ikan layang. Penelitian ini dimulai dari Februari 2023 sampai dengan April 2023 yang berlokasi di Desa Lobuto Timur Kecamatan Biluhu.

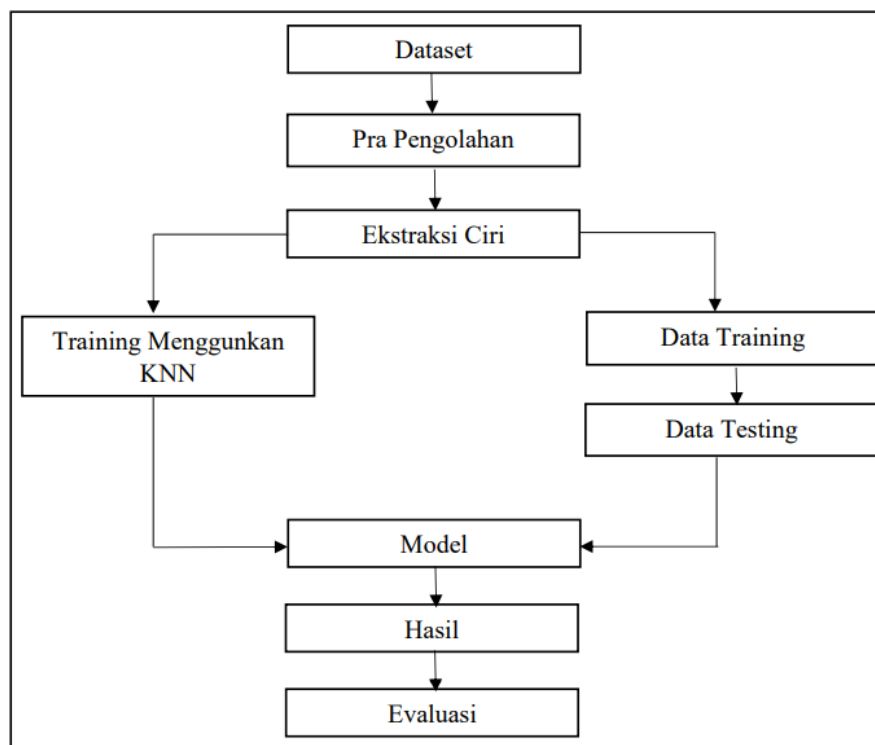


Gambar 3. 1 Desain Penelitian

3.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini terdiri dari dua jenis yaitu data primer dan data sekunder. Data primer adalah data yang di peroleh dari hasil penelitian lapangan dimana data yang dikumpulkan langsung oleh peneliti di Desa Lobuto Timur Kecamatan Biluhu. Sedangkan data sekunder adalah data yang berasal dari penelitian kepustakaan yaitu data yang sudah tersedia dan kita tinggal mencari dan mengumpulkan. Data sekunder dari penelitian ini adalah metode kepustakaan yaitu dari teori – teori yang sudah ada, berupa teori tentang pendeteksian ikan segar dan tidak segar.

3.3 Pemodelan



Gambar 3. 2 Pemodelan

3.3.1 Pra Pengolahan

Sebelum data diolah, terlebih dahulu dilakukan proses pra pengolahan yang menggunakan Histogram Equalization, hal ini dilakukan karena Histogram Equalization berfungsi untuk mengubah distribusi nilai derajat keabuan pada sebuah citra, sehingga menjadi seragam. Tujuan dari Histogram Equalization

untuk melihat apakah distribusi informasi yang ada dalam suatu citra sudah baik atau belum baik.

3.3.2 Ekstraksi Ciri

Pada proses ekstraksi ciri data input akan dibaca dan kemudian diolah dalam dua proses yang berlainan. Proses ekstraksi fitur tekstur menggunakan metode GLCM (Gray Level Co-Occurrence Matrix). Masukan dari proses ini adalah citra kualitas ikan layang yang sudah dipotong dengan ukuran tertentu. Semua citra diambil pada kondisi pencahayaan yang terang. Output dari proses ini adalah matriks $N \times 1$ dimana N adalah jumlah vektor ciri.

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 sampel ikan layang dari kualitas yang berbeda. Algoritma ini akan menentukan atau mencari bobot yang terbaik. Data training ini berupa hasil dari ekstraksi GLCM (Gray Level Co-Occurrence Matrix) terhadap kualitas pada ikan layang.

3.3.1 Training Menggunakan KNN

Training menggunakan K-NN menjadikan data training yang berupa angka atau parameter hasil dari citra kualitas ikan layang yang telah diolah melalui fitur ekstraksi GLCM sebagai data inputan untuk algoritma K-NN. Algoritma K-NN bekerja berdasarkan jarak terdekat dari training dan test dalam penentuannya, kemudian mengambil mayoritas hasil ketentuan kualitas ikan layang yang telah didapatkan untuk dijadikan prediksi dari test.

3.3.2 Data Testing

Data testing berupa data yang telah terekstraksi cirinya, untuk menguji data yang telah dilatih, jumlah data training yang digunakan berupa 90% dari sampel kualitas ikan layang yang berbeda. Data testing digunakan untuk mengetahui keberhasilan Classifier berhasil melakukan klasifikasi, data testing

merupakan ciri dari hasil ekstraksi GLCM (Gray Level Co-Occurrence Matrix) terhadap kualitas ikan layang.

3.3.3 Model

Model merupakan hasil dari proses training algoritma K-NN menggunakan data testing.

3.3.4 Hasil Klasifikasi

Hasil klasifikasi merupakan output pada data testing yang didapatkan dari proses klasifikasi yang menggunakan algoritma K-Nearest Neighbor berdasarkan model yang diperoleh dari data training.

3.3.5 Evaluasi

Evaluasi bertujuan untuk mengetahui kinerja dari metode GLCM (Gray Level Co-Occurrence Matrix) terhadap kualitas ikan layang yang digunakan. Proses evaluasi dilakukan pada seluruh data testing kemudian hasil output yang akan dipetakan kedalam Confusion Matrix untuk dihitung nilai akurasinya.

BAB IV

HASIL PENELITIAN

4.1 Hasil Pengumpulan Data

Data yang di gunakan pada penelitian ini adalah kumpulan beberapa foto daging ikan layang yang segar dan tidak segar. Di mana peneliti mengambil gambar secara langsung menggunakan kamera Smartphone Poco M4 Pro dengan resolusi kamera 64 MP dari berbagai macam angle yang berbeda. Jumlah data yang di gunakan yaitu 321 citra daging ikan layang yang terdiri dari 62 data testing dan 259 data training dengan tingkat kesegaran yang berbeda (segar 140 dan tidak segar 119).

Tabel 4. 1 Dataset Ikan Layang

SEGAR	TIDAK SEGAR
	
	

4.2 Hasil Pemodelan

4.2.1 Pra – Pengolahan

Pra – pengolahan citra yaitu proses paling awal dalam pengolahan citra sebelum proses utama di lakukan. Pada tahap ini citra ikan layang akan di konversi

agar di peroleh citra ikan layang sesuai kebutuhan. Tahap ini diperlukan agar citra dapat di normalisasi dari permasalahan luminasi yang terlalu gelap atau terlalu terang sehingga dapat meningkatkan prforma dari sistem pengenalan citra ikan segar . Pra – pengolahan dalam dalam penelitian ini dilakukan dengan 4 tahap, yaitu :

1. Pengubahan citra warna ke *grayscale*

Pada tahap ini citra *training* dan citra *testing* yang awalnya berbentuk citra RGB (red, green, blue) dirubah menjadi citra bentuk *grayscale*. Perubahan ini untuk mengurangi kebutuhan memory karena *grayscale* memiliki persamaan yang sederhana dimana nilai dari warna putih diwakili dengan angka 255 dan nilai warna hitam diwakili dengan angka 0.

2. Normalisasi Citra

Pada tahap ini citra akan di normalisasi dengan histogram ekualisasi setelah melalui tahap pengubahan citra *grayscale*.

Histogram ekualisasi adalah proses perataan histogram, dimana distribusi nilai derajat keabuan dibuat merata.

3. Segmentasi Citra

Proses segmentasi citra ini bertujuan untuk memisahkan objek yang diteliti dalam hal ini gambar ikan dengan bagian background atau gambar latar belakang dari suatu citra. Adapun pendekatan yang digunakan dalam proses segmentasi adalah dengan cara melakukan proses thresholding pada citra hasil proses peningkatan kualitas gambar

4. Metode *Otsu Thresholding*

Otsu thresholding merupakan metode segmentasi yang sederhana dalam teknik segmentasi sehingga dapat lebih mudah dalam melakukan pembagian wilayah – wilayah berdasarkan kriteria keserupaan untuk

mengenali objek. Metode otsu bertujuan untuk membagi histogram citra keabuan kedalam dua daerah yang berbeda secara otomatis.

4.2.2 Ekstraksi Fitur

Fitur ekstraksi Grey Level Co-occurrence matrix (GLCM) merupakan matrix yang menggambarkan frekuensi munculnya pasangan dua pixel dengan intensitas tertentu dalam jarak dapat berupa nilai 1,2,3,4,5 dan orientasi arah sudut dapat berupa 0° , 45° , 90° , 135° .

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

Nilai pada pada tabel 4.1 berupa matrix 3 x 3 yang sebenarnya matrix 104×104 yang di sederhanakan untuk memudahkan perhitungan.

Tabel 4. 2 Nilai Dari Suatu Citra

i/j	0	1	2
0	162	162	162
1	170	166	162
2	153	153	155

Setelah menentukan arahnya selanjutnya membentuk martiks kookurens dengan cara menghitung frekuensi kemunculan pasangan nilai keabuan piksel referensi dan peksel tetangga pada jarak dan arah yang di tentukan.Perhitungan menggunakan nilai orientasi dengan arah sudut 0° . Hasilnya sebagai berikut :

Tabel 4. 3 Matrix GLCM

i/j	153	155	162	166	170
153	1	1	0	0	0
155	0	0	0	0	0
162	0	0	2	0	0
166	0	0	1	0	0
170	0	0	0	1	0

Jumlah total nilai GLCM = 6.

Selanjutnya yaitu menormalisasikan matrix dengan cara menjumlahkan seluruh nilai yang ada dalam matrix tersebut.

Tabel 4. 4 Matrix Ternormalisasi

i/j	153	155	162	166	170
153	0,167	0,167	0	0	0
155	0	0	0	0	0
162	0	0	0,333	0	0
166	0	0	0,167	0	0
170	0	0	0	0,167	0

Setelah itu menghitung ciri statistic GLCM :

a. Menghitung nilai *Contras*

$$\text{Contras} = ((153-153)^2(0,167) + (155-153)^2(0,167) + (162-162)^2(0,333) + (162-166)^2(0,167 + (166-170)^2(0,167))$$

$$\text{Contras} = 0 + 0,67 + 0 + 1,34 + 0$$

$$\text{Contras} = \mathbf{3,34}$$

b. Menghitung nilai *Correlation*

$$\text{Correlation} =$$

$$\mu_i = (153 * (0,167)) + (155 * (0,167)) + 162 * (0,333)) + (162 * (0,167)) + (166 * (0,167))$$

$$\mu_i = 25,55 + 25,88 + 53,94 + 27,72$$

$$\mu_i = \mathbf{160,14}$$

$$\mu_j = (153 * (0,167)) + (153 * (0,167)) + 162 * (0,333)) + (166 * (0,167)) + (170 * (0,167))$$

$$\mu_j = 25,55 + 25,55 + 53,95 + 27,72 + 28,39$$

$$\mu_j = \mathbf{161,16}$$

$$\sigma_i = \sqrt{\begin{aligned} &((0,167(153 - 160,14)^2) + ((0,167(155 - 160,14)^2) \\ &+ ((0,333(162 - 160,14)^2) + ((0,167(162 - 160,14)^2) \\ &+ ((0,167(166 - 160,14)^2) \end{aligned}}$$

$$\sigma_i = \sqrt{\begin{aligned} &((0,167(-7,14)^2) + ((0,167(-5,14)^2) + \\ &((0,333(1,86)^2) + ((0,167(1,86)^2) + \\ &((0,167(5,86)^2) \end{aligned}}$$

$$\sigma_i = \mathbf{14,787}$$

$$\sigma_j = \sqrt{\begin{aligned} &((0,167(153 - 161,16)^2) + ((0,167(153 - 161,16)^2) \\ &+ ((0,333(162 - 161,16)^2) + ((0,167(166 - 161,16)^2) \\ &+ ((0,167(170 - 161,16)^2) \end{aligned}}$$

$$\sigma_j = \sqrt{\begin{aligned} &((0,167(-8,16)^2) + ((0,167(-8,16)^2) + \\ &((0,333(0,84)^2) + ((0,167(4,84)^2) + \\ &((0,167(8,84)^2) \end{aligned}}$$

$$\sigma_j = \mathbf{31,644}$$

$$\begin{aligned} \text{Correlation} = &((153 - 160,14) * (153 - 161,16) * (0,167) / (14,787 * 31,644)) + \\ &((155 - 160,14) * (153 - 161,16) * (0,167) / (14,787 * 31,644)) + ((162 - 160,14) \\ &* (162 - 161,16) * (0,333) / (14,787 * 31,644)) + ((162 - 160,14) * (166 - \\ &161,16) * (0,167) / (14,787 * 31,644)) + ((166 - 160,14) * (170 - 161,16) * \\ &(0,167) / (14,787 * 31,644)) \end{aligned}$$

$$\text{Correlation} = \mathbf{0,054}$$

c. Menghitung nilai *Energy*

$$\text{Energy} = ((0,167)^2) + (0,167)^2 + ((0,333)^2) + ((0,167)^2) + ((0,167)^2)$$

$$\text{Energy} = \mathbf{0,23}$$

- d. Menghitung nilai *Homogeneity*

$$\begin{aligned} \text{Homogeneity} &= ((0,167) / 1 + (153-153)) + ((0,167) / 1 + (155-153)) + ((0,333) \\ &/ 1 + (162-162)) + ((0,167) / 1 + (162-166)) + ((0,167) / 1 + (166-170)) \\ \text{Homogeneity} &= \mathbf{0,44} \end{aligned}$$

- e. Menghitung nilai *Entropy*

$$\begin{aligned} \text{Entropy} &= (0,167.\log 0,167) + (0,167.\log 0,167) + (0,333.\log 0,333) + \\ &(0,167.\log 0,167) + (0,167.\log 0,167) \\ \text{Entropy} &= \mathbf{0,675} \end{aligned}$$

4.2.3 Klasifikasi

Tabel 4. 5 Hasil Training Data Set

Data Training	X1	X2	X3	X4	X5	Kelas
0	1.051626	-1.23425	0.931347	1.433605	0.136068	Segar
1	0.02723	-0.22025	0.073198	0.147108	-0.60069	TidakSegar
2	1.51677	-1.38109	1.299692	1.849845	0.162546	Segar
3	0.340814	-1.06527	0.343385	0.811305	0.945806	Segar
...	...	...	...	...	...	...
58	0.446304	1.411297	0.432718	0.161886	-0.46683	Segar
59	-0.05377	-1.03709	0.00223	0.017269	0.552719	TidakSegar
60	0.539525	-1.21956	0.511035	0.733485	-0.65531	Segar
61	0.124275	-0.63463	0.157574	-0.08089	1.099108	TidakSegar
Data Testing	0.087215	1.633701	0.125434	-0.29007	-0.31547	?

Untuk menentukan kelas dari data uji maka terlebih dahulu ditentukan nilai $K=5$. Setelah itu setiap jarak dari data latih ke data uji di hitung menggunakan rumus perhitungan jarak euclidean distance seperti dibawah ini:

$$\begin{aligned} d = \sqrt{(1,0516 - 0,0872)^2 + (-1,2342 - 1,6337)^2 + (0,9313 - 0,1254)^2 + \\ (1,4336 - (-0,2900))^2 + (0,1360 - (-0,3154))^2} \end{aligned}$$

$$\begin{aligned}
&= \sqrt{(0,9644)^2 + (-2,8679)^2 + (0,8059)^2 + (1,7236)^2 + (0,4514)^2} \\
&= \\
&= \sqrt{0,93006376 + (8,22485041 + (0,64947481 + (2,97079696 + (0,20376196 \\
&= \sqrt{12,9789479} = 3,60263069
\end{aligned}$$

Tabel 4. 6 Perhitungan Jarak Euclidean Distance

Data Training	Perhitungan jarak <i>Euclidean Distance</i>	Nilai
0	$d = \sqrt{(1,0516 - 0,0872)^2 + (-1,2342 - 1,6337)^2 + (0,9313 - 0,1254)^2 + (1,4336 - (-0,2900))^2 + (0,1360 - (-0,3154))^2}$	3,60263069
1	$d = \sqrt{(0,0272 - 0,0872)^2 + (-0,2202 - 1,6337)^2 + (0,0731 - 0,1254)^2 + (0,1471 - (-0,2900))^2 + (0,6006 - (-0,3154))^2}$	2,12268531
2	$d = \sqrt{(1,5167 - 0,0872)^2 + (-1,3810 - 1,6337)^2 + (1,2996 - 0,1254)^2 + (1,8498 - (-0,2900))^2 + (0,1625 - (-0,3154))^2}$	4,16146181
3	$d = \sqrt{(0,3408 - 0,0872)^2 + (-1,0652 - 1,6337)^2 + (0,3433 - 0,1254)^2 + (0,8113 - (-0,2900))^2 + (0,9458 - (-0,3154))^2}$	3,19364082
...	...	...
58	$d = \sqrt{(0,4463 - 0,0872)^2 + (-1,4112 - 1,6337)^2 + (0,4327 - 0,1254)^2 + (0,1618 - (-0,2900))^2 + (-0,4668 - (-0,3154))^2}$	3,1179877
59	$d = \sqrt{(-0,053 - 0,0872)^2 + (-1,0370 - 1,6337)^2 + (0,0022 - 0,1254)^2 + (0,0172 - (-0,2900))^2 + (0,5527 - (-0,3154))^2}$	2,83119038
60	$d = \sqrt{(0,1242 - 0,0872)^2 + (-1,2195 - 1,6337)^2 + (0,5110 - 0,1254)^2 + (0,7334 - (-0,2900))^2 + (-0,6553 - (-0,3154))^2}$	3,07468473
61	$d = \sqrt{(0,1242 - 0,0872)^2 + (-0,6346 - 1,6337)^2 + (0,1575 - 0,1254)^2 + (-0,0808 - (-0,2900))^2 + (1,0991 - (-0,3154))^2}$	2,68182013

Hasil yang didapatkan dari perhitungan jarak *Euclidean distance* untuk data uji di atas adalah data testing segar.

4.2.1 Evaluasi

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 62 data uji dengan jumlah $k=5$ sebagai berikut :

Tabel 4. 7 Hasil Data Testing

Data	Target	Aktual	Data	Target	Aktual
Data 0	Segar	Segar	Data 32	Tidak Segar	Tidak Segar
Data 1	Tidak Segar	Tidak Segar	Data 33	Segar	Segar
Data 2	Segar	Segar	Data 34	Segar	Segar
Data 3	Tidak Segar	Segar	Data 35	Segar	Segar
Data 4	Tidak Segar	Tidak Segar	Data 36	Segar	Segar
Data 5	Segar	Tidak Segar	Data 37	Segar	Segar
Data 6	Segar	Segar	Data 38	Segar	Segar
Data 7	Segar	Segar	Data 39	Tidak Segar	Tidak Segar
Data 8	Segar	Segar	Data 40	Segar	Tidak Segar
Data 9	Tidak Segar	Tidak Segar	Data 41	Segar	Segar
Data 10	Tidak Segar	Segar	Data 42	Tidak Segar	Tidak Segar
Data 11	Segar	Segar	Data 43	Tidak Segar	Tidak Segar
Data 12	Segar	Segar	Data 44	Tidak Segar	Tidak Segar
Data 13	Segar	Segar	Data 45	Segar	Segar
Data 14	Tidak Segar	Tidak Segar	Data 46	Tidak Segar	Tidak Segar
Data 15	Tidak Segar	Tidak Segar	Data 47	Segar	Segar
Data 16	Tidak Segar	Tidak Segar	Data 48	Segar	Segar
Data 17	Segar	Segar	Data 49	Tidak Segar	Tidak Segar
Data 18	Segar	Segar	Data 50	Tidak Segar	Tidak Segar
Data 19	Segar	Segar	Data 51	Segar	Segar
Data 20	Segar	Segar	Data 52	Segar	Segar
Data 21	Tidak Segar	Tidak Segar	Data 53	Tidak Segar	Tidak Segar
Data 22	Tidak Segar	Tidak Segar	Data 54	Segar	Tidak Segar
Data 23	Tidak Segar	Tidak Segar	Data 55	Tidak Segar	Tidak Segar
Data 24	Segar	Segar	Data 56	Segar	Tidak Segar
Data 25	Segar	Segar	Data 57	Tidak Segar	Tidak Segar

Data 26	Tidak Segar	Tidak Segar	Data 58	Segar	Segar
Data 27	Segar	Tidak Segar	Data 59	Tidak Segar	Tidak Segar
Data 28	Segar	Segar	Data 60	Segar	Segar
Data 29	Segar	Tidak Segar	Data 61	Segar	Tidak Segar
Data 30	Tidak Segar	Tidak Segar			
Data 31	Tidak Segar	Tidak Segar			

Tabel 4. 8 Hasil Data Uji K=5

Kelas	Segar	Tidak Segar
Segar	29	7
Tidak Segar	2	24

Jumlah data yang di klasifikasikan benar = 53, jumlah data yang di klasifikasikan salah = 9.

Keterangan :

(TP) *True Positive* = 29

(TN) *True Negative* = 24

(FP) *False Positive* = 2

(FN) *False Negative* = 7

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

$$Akurasi = \frac{53}{62} \times 100\% = 85,5\%$$

$$Presisi = \frac{TP}{TP + FP} \times 100\%$$

$$Presisi = \frac{29}{31} \times 100\% = 93,5\%$$

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

$$Recall = \frac{29}{36} \times 100\% = 80,5\%$$

BAB V

PEMBAHASAN

5.1 Pembahasan Model

Berdasarkan perolehan hasil percobaan yang telah dilakukan pada bab sebelumnya, maka dapat dijelaskan beberapa hal sebagai berikut :

- Data yang berhasil dikumpulkan pada penelitian ini adalah sebanyak 321 citra daging ikan layang, yang di bagi kedalam dua kategori kelas yakni segar dan tidak segar. Untuk citra ikan segar yaitu ikan yang didapat langsung dari pelelangan. Sedangkan untuk citra ikan tidak segar yaitu ikan yang disimpan selama bebrapa hari dalam peti pendingin berisi es.
- Percobaan yang dilakukan pada penelitian ini fitur citra di ekstraksi menggunakan metode Gray Level Co-occurrence Matrix dengan menghasilkan lima fitur statistik dari citra, sehingga dapat mengurangi waktu komputasi dari algoritma klasifikasi yang digunakan pada penelitian ini. Penggunaan algoritma K-NN ini menjadi lebih efisien untuk di terapkan pada aplikasi yang akan dikembangkan kedepannya.
- Dalam penggunaan metode GLCM untuk ekstraksi fitur ini memanfaatkan lima nilai fitur statistik sebagai hasil ekstraksi fitur di antaranya nilai Contrasts, Energy, Homogeneity, Angular Second Moment (ASM) dan Correlation. Kelima fitur statistik tersebut akan bertindak sebagai nilai yang mewakili setiap citra untuk kemudian menjadi variabel input pada algoritma klasifikasi yang digunakan.
- Citra daging Ikan Layang di ekstraksi ciri menggunakan *Gray Level Co-occurrence Matrix*. Setelah di dapatkan ciri dari citra daging tersebut, selanjutnya dilakukan proses klasifikasi menggunakan *K-Nearest Neighbor*. Pada tahap ini citra pada daging dijadikan sebagai inputan untuk mencari nilai akurasi terbaik pada percobaan model K-NN.

- Berdasarkan hasil percobaan klasifikasi K-NN dengan nilai K=5, menggunakan arah GLCM 0° dengan jarak 1 mendapatkan hasil akurasi sebesar 85,5%, nilai presisi sebesar 93,5%, dan recall sebesar 80,5% dengan 62 data testing.
- Berdasarkan nilai kinerja model tersebut menunjukkan bahwa penggunaan algoritma K-NN dapat bekerja dengan baik ketika digunakan bersamaan dengan metode GLCM sebagai metode ekstraksi fitur.

Tabel 5. 1 Hasil Evaluasi Nilai

Nilai	Akurasi	Presisi	Recall
K-5	85,5%	93,5%	80,5%

1.2 Pembahasan System

Google Colab akan mereset notebook beserta semua filenya maksimal 12 jam sekali. Karenanya akan lebih baik jika file yang akan kita gunakan tersimpan dengan rapi di Google Drive. Kita bisa melakukan hal tersebut dengan menjalankan program di bawah.

1. Penghubung Google Drive

```
from google.colab import drive
drive.mount('/content/drive')
```

Gambar 5.1 Penghubung Google Drive

Jika Perintah di atas dijalankan, maka kita akan diberikan URL yang akan mengantarkan kita kehalaman permohonan akses Google Drive.

2. Kode Akses Penghubung

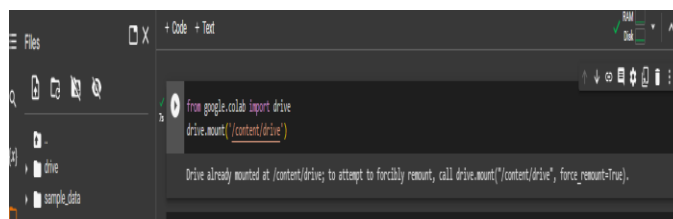
```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

Gambar 5.2 Kode Akses Penghubung

Setelah terhubung, maka akan tampak daftar file dibagian kiri notebook.

3. Folder Penghubung



The screenshot shows the Google Colab interface. On the left, there is a file explorer pane with a sidebar containing icons for files and folders. The main area of the file explorer shows a list of folders: 'drive' and 'sample_data'. The 'drive' folder is highlighted. The main code editor area on the right shows the same code as in Gambar 5.2, with the output message: 'Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).'

Gambar 5.3 Folder Penghubung

Untuk mengakses file-file tersebut kita arahkan proses load atau save ke path drive/My Drive/Folder Tujuan.

4. File Penghubung

```
[ ] from skimage import feature
import os
import cv2
from imutils import paths
import numpy as np
import pandas as pd
from skimage.filters import threshold_otsu
from matplotlib import pyplot as plt
folder_train = "/content/drive/MyDrive/Dataset/Training"
```

Gambar 5.4 File Penghubung

Pada bagian ini berfungsi untuk menyatukan file dari google collab dengan google drive yang dimana google drive ini berperan penting sebagai penyimpanan data yang sangat di butuhkan untuk menyimpan data set atau apa saja yang nanti di butuhkan dari project itu.

5. Source Code Program Deteksi

```
def ekstraksi_data_training(folder_train, radius, angle):
[ ] feature_training = []
    impaths = get_path(folder_train)
    for item in impaths:
        hasil_prapengolahan = pra_pengolahan(item)
        hasil_segmentasi = segmentasi(item)
        hasil_ekstraksi = ekstraksi_ciri(hasil_segmentasi, radius, angle)
        feature_training.append(hasil_ekstraksi)
    return feature_training

def ekstraksi_data_testing(imgtest, radius, angle):
[ ] feature_testing = []
    hasil_prapengolahan = pra_pengolahan(imgtest)
    hasil_segmentasi = segmentasi(hasil_prapengolahan)
    hasil_ekstraksi = ekstraksi_ciri(hasil_segmentasi, radius, angle)
    feature_testing.append(hasil_ekstraksi)
    return feature_testing

def getlabel(folder_train):
[ ] label = []
    impaths = get_path(folder_train)
    for item in impaths:
        label.append(os.path.split(os.path.dirname(item)) [-1])
    return label

[ ] label = getlabel(folder_train)
    data = ekstraksi_data_training(folder_train, 1, 0)

def pra_pengolahan(img_path):
[ ] img = cv2.imread(img_path)
    img = cv2.resize(img, (256, 256))
    rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    gray = cv2.cvtColor(rgb, cv2.COLOR_RGB2GRAY)
    kernel = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]])
    result = cv2.filter2D(gray, -1, kernel)
    return result, img

def segmentasi(impath):
[ ] im_preprocessing, img = pra_pengolahan(impath)
    _threshold = cv2.threshold(im_preprocessing, np.mean(im_preprocessing), 255, cv2.THRESH_BINARY_INV)
    edges = cv2.dilate(cv2.Canny(threshold, 4, 255), None)
    cnts = sorted(cv2.findContours(edges, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE) [-2], key=cv2.contourArea)[-1]
    mask = np.zeros((256, 256), np.uint8)
    masked = cv2.drawContours(mask, [cnts], -1, 255, -1)
    dst = cv2.bitwise_and(img, img, mask=mask)
    segmented = cv2.cvtColor(dst, cv2.COLOR_BGR2RGB)
    return segmented

def ekstraksi_ciri(img, radius, angle):
[ ] img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    gloc = feature.graycomatrix(img, [radius], [angle], symmetric=False, normed=True)
    aloc = feature.graycoprops(gloc, 'corr') [0,0]
    cloc = feature.graycoprops(gloc, 'contrast') [0,0]
    eloc = feature.graycoprops(gloc, 'energy') [0,0]
    img = feature.graycoprops(gloc, 'homogeneity') [0,0]
    cor = feature.graycoprops(gloc, 'correlation') [0,0]
    return [aloc, cloc, eloc, img, cor]
```

Gambar 5.5 Source Code Program Deteksi

Pada gambar di atas adalah class merupakan variabel-variabel gambar yang akan di proses di pra-pengolahan, segmentasi, ekstraksi ciri dan training.

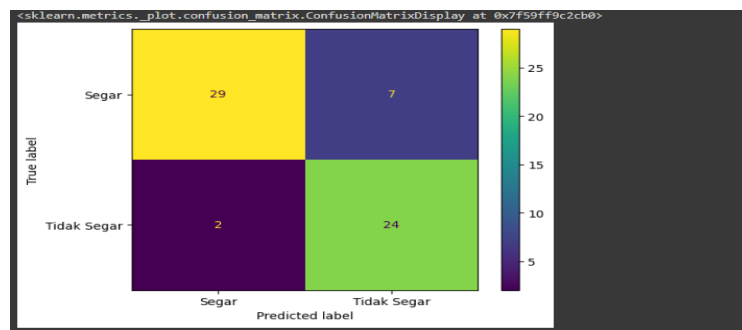
6. Tampilan Hasil Keseluruhan

```
[ ] from sklearn.metrics import accuracy_score, ConfusionMatrixDisplay, confusion_matrix
y_target = label_encoder.inverse_transform(y_val)
y_aktual = label_encoder.inverse_transform(yji_validasi_model)

cm = confusion_matrix(y_target, y_aktual, labels=['Segar', 'Tidak Segar'])
disp = ConfusionMatrixDisplay(cm, display_labels=['Segar', 'Tidak Segar'])
disp.plot()
```

Gambar 5.6 Script confusion matrix

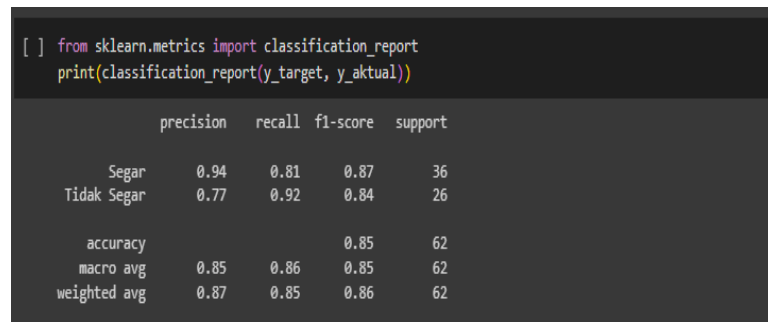
Pengujian confusion matrix bertujuan untuk evaluasi hasil dari prediksi yang di lakukan metode klasifikasi K-Nearest Neighbor. *Confusion script* untuk *confusion matrix* seperti gambar di atas dan di hasilkan *output confusion matrix* seperti pada Gambar 5.7.



Gambar 5.7 Output Confusion Matrix

Pada gambar di atas menunjukkan bahwa terdapat 24 data Tidak Segar yang benar diprediksi termasuk ke dalam kelas Tidak Segar dan 2 data Tidak Segar yang diprediksi termasuk kedalam kelas Segar. Kemudian terdapat 29 data Segar yang benar diprediksi termasuk kedalam kelas Segar dan 7 data Segar yang diprediksi termasuk kedalam kelas Tidak Segar.

Confusion matrix juga dapat digunakan untuk menghitung nilai ukuran evaluasi model yang lainnya seperti *recall*, *precision* dan *accuracy* seperti pada gambar 5.8.



```
[ ] from sklearn.metrics import classification_report
print(classification_report(y_target, y_aktual))
```

	precision	recall	f1-score	support
Segar	0.94	0.81	0.87	36
Tidak Segar	0.77	0.92	0.84	26
accuracy			0.85	62
macro avg	0.85	0.86	0.85	62
weighted avg	0.87	0.85	0.86	62

Gambar 5.8 Accuracy Training dan Testing

Berdasarkan gambar dari hasil pengujian diatas dapat diketahui bahwa nilai akurasi yang diperoleh yakni 85%. Dari hasil tersebut dapat disimpulkan bahwa semakin tinggi nilai akurasinya maka semakin baik hasil akurasi yang didapat.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil pembahasan yang telah diuraikan sebelumnya, maka pada penelitian ini dapat disimpulkan bahwa :

1. Metode ekstraksi fitur *Gray Level Co-occurrence Matrix* (GLCM) dan klasifikasi *K-Nearest Neighbor* (KNN) telah berhasil diimplementasikan pada proses pendeteksian kualitas ikan dengan tingkat akurasi 85,5% dengan nilai $k = 5$.
2. Berdasarkan hasil penelitian untuk penerapan teknik pengolahan citra dalam dunia perikanan dan pemasaran sudah bisa diterapkan dengan hasil yang cukup memuaskan dengan nilai akurasi 85,5%, presisi 93,5% dan recall 80,5%.

6.2 Saran

Dari hasil penelitian yang telah dilaksanakan maka penulis menyampaikan sejumlah untuk pengembangan penelitian ini selanjutnya diantaranya :

1. Perlu dilakukan pengumpulan data yang lebih baik lagi terutama pada jumlah data yang digunakan sehingga ada peluang untuk meningkatkan kinerja model agar dapat terimplementasi dengan baik ketika dilakukan pengembangan sistem untuk digunakan dalam dunia industry.
2. Diharapkan kedepannya agar penulis dapat menggunakan metode lain untuk mendapatkan hasil yang lebih baik.

DAFTAR PUSTAKA

- [1]. Arham, Purnama, IK.E., Purnomo, M.H., 2014, Penentuan Kualitas Kesegaran Ikan dengan Citra Mata Menggunakan Support Vector Machine, Jurnal AgriTechno, Vol. 7, No. 1, p. 10 – 19.
- [2]. ADISANJAYA, N. 2010. Potensi, Produksi Sumber Daya Ikan di Perairan Laut Indonesia dan Permasalahannya. Seminar Potensi Hasil Perikanan Indonesia. Jakarta
- [3]. INDRABAYU, NISWAR, M., AMAN, A. 2016.Sistem Pendeteksi Kesegaran Ikan Bandeng Menggunakan Citra. Jurnal Infotel Vol.8 No.2.
- [4]. Leidiyana, H., 2013, Penerapan Algoritma K-Nearest Neighbor untuk Penentuan Resiko Kredit Kepemilikan Kendaraan Bermotor, Jurnal Penelitian Ilmu Komputer, System Embedded & Logic, Vol. 1, No. 1, p. 65 – 76
- [5]. Pareza Alam Jusia, Vol, 1 No 2(2019): Jurnal Ilmiah Mahasiswa Teknik Informatika Penerapan Data Mining Untuk Clustering Data Penduduk Miskin Menggunakan Algoritma K-Means.
- [6]. Madenda, S., 2015, Pengolahan Citra dan Video Digital,. Erlangga, Jakarta.
- [7]. Clauditta, Lovidianti, Alamsyah, D., & Yohannes., —Menghitung Jumlah Orang dengan Ekstraksi Fitur Gray Level Co-occurrence Matrix (GLCM)l. STMIK GI MDP Palembang, (x), 2016.
- [8]. Maura, A.W., —Identifikasi Kematangan Buah Apel Dengan Gray Level Co-Occurrence Matrix (GLCM), Teknik Informatika Komputer, (114), 2017.
- [9]. MURNIATI, A.S & SUNARMAN. 2000.Pendinginan, Pembekuan, Pengawetan Ikan. Kanisius. Yogyakarta.
- [10]. Asroni, Ronald Adrian, Mey 2015 Penerapan Metode K-Means Untuk Clustering Mahasiswa Berdasarkan Nilai Akademik

- [11]. Fitri Astutik, “Sistem Pengenalan Kualitas Ikan Gurame dengan Wavelet, PCA, Histogram HSV dan KNN” Lontar Komputer Vol. 4, No. 2 Desember 2013.
- [12]. Nikolaus Situmorang, “Penerapan Metode K-Nearest Neighbor dalam Identifikasi Kesegaran Ikan” Skripsi, Medan 2019.
- [13]. Rio Pradhitya, “Pembangunan Aplikasi Deteksi dan Tracking Warna Virtual Drawing Menggunakan Algoritma Color Filtering”, Universitas Komputer Indonesia.
- [14]. E. Miyamoto and T. M. Jr., “FAST CALCULATION OF HARALICK TEXTURE FEATURES Human Computer Interaction Institute Department of Electrical and Computer Engineering Carnegie Mellon University,” Hum. Comput. Interact. Inst. Dep. Electr. Comput. Eng. Carnegie Mellon Univ. Pittsburgh PA, vol. 15213, pp. 1–6, 2011.
- [15]. R. Kusumanto and A. . N. Tompunu, “Technogenic activity of man and local sources of environmental pollution,” Pengolah. CITRA Digit. UNTUK MENDETEKSI OBYEK MENGGUNAKAN Pengolah. Warn. Model Norm. RGB, vol. (Semantik, no. ISBN 979-26-0255-0, 2011.
- [16]. Y. Fernando, “Klasifikasi Jenis Daging Berdasarkan Analisis Citra Tekstur Gray Level Co-Occurrence Matrices (Glcm) Dan Warna,” Semin. Nas. Sains dan Teknol. 2017, no. Fakultas Teknik Universitas Muhammadiyah Jakarta, pp. 1–7, 2017.
- [17]. E. D. Handoyo, “Perancangan Mini Image Editor Versi 1.0 sebagai Aplikasi Penunjang Mata Kuliah Digital Image Processing,” J. Inform., vol. 2, no. 2, pp. 145–154, 2006.
- [18]. C. N. Santi, “Mengubah Citra Berwarna Menjadi Gray-Scale dan Citra biner,” Teknol. Inf. Din., vol. 16, no. 1, pp. 14–19, 2011.
- [19]. Agus Wahyu Sanjaya, Achmad Rizal dan Inung Wijayanto, “Deteksi Penyakit Kulit Menggunakan Analisis Fitur Warna Dan Tekstur Dengan Metode Color Moment, Gray Level Cooccurence Matrix, Dan Jaringan Saraf Tiruan Backpropagaton” Fakultas Teknik Elektro, Universitas Telkom.

- [20]. Restu Widodo, Agus Wahyu Widodo, Arry Supriyanto “Pemanfaatan Ciri Gray Level Co-Occurrence Matrix (GLCM) Citra Buah Jeruk Keprok (*Citrus reticulata* Blanco) untuk Klasifikasi Mutu” Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 2, No. 11, November 2018.
- [21]. Eleyan, Demirel, “Penerapan Metode K-Nearest Neighbor dalam Identifikasi Kesegaran Ikan” Skripsi, Medan 2019.
- [22]. Kadir, Susanto, “Penerapan Metode K-Nearest Neighbor dalam Identifikasi Kesegaran Ikan” Skripsi, Medan 2019.
- [23]. Ni Luh Gede Pivin Suwirmayanti, “Penerapan Metode K-Nearest Neighbor Untuk Sistem Rekomendasi Pemilihan Mobil” Techno.COM, Vol. 16, No. 2, Mei 2017.
- [24]. Aris, “Penerapan Algoritma Grey Level Co-Occurrence Matrix (GlcM) Dengan Metode K-NearestNeighbour Untuk Klasifikasi Beras Berdasarkan Tekstur” Skripsi, 2015.
- [25]. Widya Eka Wardani, "Pengenalan Motif Batik Menggunakan Metode Transformer Paket Wavelet," Tugas Akhir Teknik Informatika, no.Universitas Widyatama, Bandung, 2013.

Lampiran

Lampiran 1. 1 Kode Program

```
from google.colab import drive
drive.mount('/content/drive')
```

```
from skimage import feature
import os
import cv2
from imutils import paths
import numpy as np
import pandas as pd
from skimage.filters import threshold_otsu
from matplotlib import pyplot as plt
folder_train = "/content/drive/MyDrive/Dataset/Training"
```

```
def get_path(folder_train):
    array_path = []
    for path in paths.list_images(folder_train):
        array_path.append(path)
    return array_path
```

```
def pra_pengolahan(img_path):
    img = cv2.imread(img_path)
    img = cv2.resize(img, (256, 256))
    rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    gray = cv2.cvtColor(rgb, cv2.COLOR_RGB2GRAY)
    kernel = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]])
    result = cv2.filter2D(gray, -1, kernel)
    return result, img
```

```
def segmentasi(impath):
    im_preprocessing, img = pra_pengolahan(impath)
    _, thresh = cv2.threshold(im_preprocessing,
np.mean(im_preprocessing), 255, cv2.THRESH_BINARY_INV)
    edges = cv2.dilate(cv2.Canny(thresh, 0, 255), None)
```

```

    cnt = sorted(cv2.findContours(edges, cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE) [-2], key=cv2.contourArea)[-1]
    mask = np.zeros((256,256), np.uint8)
    masked = cv2.drawContours(mask, [cnt], -1, 255, -1)
    dst = cv2.bitwise_and(img, img, mask=mask)
    segmented = cv2.cvtColor(dst, cv2.COLOR_BGR2RGB)
    return segmented

```

```

def ekstraksi_ciri(img, radius, angle):
    img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
    glcm = feature.graycomatrix(img, [radius], [angle],
symmetric=False, normed=True)
    asm = feature.graycoprops(glcm, 'ASM') [0,0]
    con = feature.graycoprops(glcm, 'contrast') [0,0]
    ene = feature.graycoprops(glcm, 'energy') [0,0]
    hmg = feature.graycoprops(glcm, 'homogeneity') [0,0]
    cor = feature.graycoprops(glcm, 'correlation') [0,0]
    return [asm, con, ene, hmg, cor]

```

```

def ekstraksi_data_training(folder_train, radius, angle):
    feature_training = []
    impaths = get_path(folder_train)
    for item in impaths:
        hasil_prapengolahan = pra_pengolahan(item)
        hasil_segmentasi = segmentasi(item)
        hasil_ekstraksi = ekstraksi_ciri(hasil_segmentasi,
radius, angle)
        feature_training.append(hasil_ekstraksi)
    return feature_training

```

```

def ekstraksi_data_testing(imgtest, radius, angle):
    feature_testing = []
    hasil_prapengolahan = pra_pengolahan(imgtest)
    hasil_segmentasi = segmentasi(hasil_prapengolahan)
    hasil_ekstraksi = ekstraksi_ciri(hasil_segmentasi, radius,
angle)
    feature_testing.append(hasil_ekstraksi)
    return feature_testing

```

```

def getLabel(folder_train):
    label = []
    impaths = get_path(folder_train)

```

```

for item in impaths:
    label.append(os.path.split(os.path.dirname(item)))[-1])
return label

```

```

label = getLabel(folder_train)
data = ekstraksi_data_training(folder_train, 1, 0)

```

```

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler

```

```

label_encoder = LabelEncoder()
y = label_encoder.fit_transform(label)

normalisasi = StandardScaler()
x = normalisasi.fit_transform(data)

```

```

x_train, x_val, y_train, y_val = train_test_split(x, y,
test_size=0.2, random_state=4)

```

```

from sklearn.neighbors import KNeighborsClassifier

model = KNeighborsClassifier(n_neighbors = 5)
model.fit(x_train, y_train)

uji_validasi_model = model.predict(x_val)

```

```

from sklearn.metrics import accuracy_score,
ConfusionMatrixDisplay, confusion_matrix
y_target = label_encoder.inverse_transform(y_val)
y_aktual = label_encoder.inverse_transform(uji_validasi_model)

cm = confusion_matrix(y_target, y_aktual, labels=['Segar',
'Tidak Segar'])
disp = ConfusionMatrixDisplay(cm, display_labels=['Segar',
'Tidak Segar'])

```

```

from sklearn.metrics import classification_report
print(classification_report(y_target, y_aktual))

```

Lampiran 1. 2 Surat Rekomendasi Penelitian



PEMERINTAH KABUPATEN GORONTALO
KECAMATAN BILUHU
DESA LOBUTO TIMUR
Jln Buke Panai, Kode Pos 96272

SURAT KETERANGAN PENELITIAN

Yang Bertanda Tangan dibawah ini :

Nama : Arifin Abdulkadir
Jabatan : Kepala Desa Lobuto Timur
Alamat : Desa Lobuto Timur

Menerangkan Kepada

Nama : Siti Regina Khairunniza A. Ahmad
NIM : T3116064
Mahasiswa : Universitas Ichsan Gorontalo
Program Studi : Teknik Informatika

Dengan ini menerangkan bahwa yang bersangkutan benar-benar telah melakukan penelitian tentang "Deteksi Kesegaran Ikan Layang Berdasarkan Citra Daging Menggunakan Metode GLCM dan K-NN." Guna untuk menyelesaikan Studi pada Program Studi Teknik Informatika Fakultas Ilmu Komputer, dan bersangkutan telah menyelesaikan penelitian tersebut pada tanggal 10 April 2023 sesuai dengan waktu yang telah ditentukan.

Demikian Surat Keterangan ini dibuat untuk dapat dipergunakan seperlunya.

Lobuto Timur, 10 April 2023

KEPALA DESA


ARIFIN ABDULKADIR



Dipindai dengan CamScanner

Lampiran 1.3 Hasil Turnitin


Similarity Report ID: oid:25211:35561957

● **27% Overall Similarity**

Top sources found in the following databases:

- 27% Internet database
- 6% Publications database
- Crossref database
- Crossref Posted Content database
- 0% Submitted Works database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	text-id.123dok.com Internet	11%
2	cosphijournal.unisan.ac.id Internet	3%
3	cosphijournal.unisan.ac.id Internet	2%
4	repositori.usu.ac.id Internet	1%
5	researchgate.net Internet	1%
6	media.neliti.com Internet	1%
7	ojs.umrah.ac.id Internet	1%
8	id.123dok.com Internet	1%

Sources overview


Similarity Report ID: oid:25211:35561957

9	es.scribd.com Internet	1%
10	eprints.unram.ac.id Internet	<1%
11	repository.uin-suska.ac.id Internet	<1%
12	eprints.unisla.ac.id Internet	<1%
13	andi.ddns.net Internet	<1%
14	ejournal.stipwunaraha.ac.id Internet	<1%
15	ojs.selodangmayang.com Internet	<1%

Lampiran 1. 4 Riwayat Hidup



DAFTAR RIWAYAT HIDUP

Nama : Siti Regina Khairunniza A. Ahmad
Tempat, Tgl Lahir : Kab.Gorontalo, 05 Mei 1998
Agama : Islam
Email : reginaahmad55@gmail.com

Riwayat Pendidikan :

1. Tahun 2011, menyelesaikan Pendidikan di Sekolah Dasar Negeri 1 Biluhu
2. Tahun 2013, menyelesaikan Pendidikan di Sekolah Menengah Pertama Negeri 1 Biluhu
3. Tahun 2016, menyelesaikan Pendidikan di Sekolah Menengah Atas Negeri 1 Biluhu
4. Tahun 2016, telah diterima menjadi Mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo