

**DETEKSI KUALITAS IKAN CAKALANG
MENGUNAKAN METODE *GLCM*
DAN ALGORITMA *KNN***
(STUDI KASUS : DINAS KELAUTAN DAN PERIKANAN
PROVINSI GORONTALO)

Oleh

HAMDAN PAKAYA

T3116102

SKRIPSI

Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar Sarjana



**PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2020**

**DETEKSI KUALITAS IKAN CAKALANG
MENGUNAKAN METODE *GLCM*
DAN ALGORITMA *KNN***
(STUDI KASUS : DINAS KELAUTAN DAN PERIKANAN
PROVINSI GORONTALO)

Oleh

HAMDAN PAKAYA

T3116102

SKRIPSI

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



**PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2020**

PENGESAHAN SKRIPSI
DETEKSI KUALITAS IKAN CAKALANG
MENGGUNAKAN METODE *GLCM*
DAN ALGORITMA *KNN*

Oleh

HAMDAN PAKAYAN

T3116102

SKRIPSI

Untuk memenuhi salah satu syarat ujian guna memperoleh gelar Sarjana Program
Studi Teknik Informatika, ini telah disetujui oleh Tim Pembimbing

Gorontalo, 14 Juli 2020

Pembimbing I

Pembimbing II



Sudirman S.Panna, M.Kom
NIDN. 9909912852



Kartika Chandra Pelangi, M.Kom
NIDN. 0916038304

PERSETUJUAN SKRIPSI

DETEKSI KUALITAS IKAN CAKALANG MENGGUNAKAN METODE *GLCM* DAN ALGORITMA *KNN*

Oleh
HAMDAN PAKAYA

T3116102

Di Periksa oleh Panitia Ujian Strata Satu (S1)
Universitas Ichsan Gorontalo
Gorontalo, 14 Juli 2020

1. Ketua Penguji
Irvan A.Salihi, M.Kom
2. Anggota Penguji
Abd.Rahmat Karim Haba, M. Kom
3. Anggota Penguji
Serwin, M.Kom
4. Anggota Penguji
Sudirman S.Panna, M.Kom
5. Anggota Penguji
Kartika Chandra Pelangi, M.Kom

Mengetahui :


Dekan Fakultas Ilmu Komputer
Zohrahayaty, M.Kom
NIDN. 0912117702

Ketua Program Studi


Irvan Abraham Salihi, M.Kom
NIDN. 0928028101

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 dari arahan Tim Pembimbing.
3. Dalam karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah dipublikasi orang lain, kecuali serta tertulis dicantumkan sebagai acuan/ sitasi dalam naskah dan dicantumkan pula dalam daftar pustaka.
4. Pernyataan isi saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidak benaran 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, 14 Juli 2020

Yang Membuat Pernyataan



Hamdan Pakaya

ABSTRACT

Detection of quality in this study was divided into two groups namely Fresh, Rotten. This study aims to classify freshness based on the level of accuracy using the feature extraction method that is the gray level co-occurrence matrix. This research method consists of: conversion of RGB data to grayscale, image normalization, detection of skipjack tuna quality using color texture, feature extraction and classification. In this study used training data as many as 90 skipjack fish images which are real data. The data consists of the quality of skipjack tuna above and each of them consists of Fresh 45, and Rot 45. The image data is processed into grayscale image which is then detected by the quality of skipjack tuna using color texture. After obtaining the quality of skipjack fish, then the characteristics are calculated using the gray level co-occurrence matrix. The algorithm used for the classification process is the k-nearest neighbor algorithm. The final results of the test show that the proposed method has been able to detect the quality of skipjack tuna using color texture with the results of accuracy calculated using a confusion matrix, it can be concluded that in the K-NN model experiments, with the number of $K = 1$, and GLCM Direction = 0o , with a distance = 1, then get the highest accuracy value = 70%. Whereas in the K-NN model experiments, with the number of $K = 2$, and GLCM Direction = 135o, with a distance = 2, the highest accuracy value = 60% is obtained. In the K-NN model experiment, with the number of $K = 3$, and GLCM Direction = 135o, with a distance = 1, then get an Accuracy Value of 60%. In the K-NN model experiment, with the number of $K = 4$, and GLCM Direction = 90o, with a distance = 4, then get an Accuracy Value = 80%. In experiments with the number of $K = 4$, and GLCM direction = 45o, with a distance = 4, then get the highest accuracy value = 50%. While in experiments with the number of $K = 3$, and GLCM Direction = 45o, with a distance = 3, the highest accuracy value = 70% is obtained. And with the number of $K = 1$, and GLCM = 45o Direction, with a distance = 1, then get an Accuracy Value = 50%. In the K-NN model experiment, with the number of $K = 2$, and GLCM Direction = 90o, with a distance = 2, then get an Accuracy Value of 40%. Thus the application of the gray level co-occurrence matrix and k-nearest neighbor methods on the problem of detecting the quality of skipjack fish using color texture needs to be developed again accurately.

Keywords: Quality Detection, Gray Level Co-Occurrence Matrix, K-Nearest Neighbor

ABSTRAK

Deteksi kualitas dalam penelitian ini dibagi menjadi dua kelompok yaitu Segar, Busuk. Penelitian ini bertujuan untuk mengklasifikasi kesegaran berdasarkan tingkat akurasi dengan menggunakan metode fitur ekstraksi yaitu gray level co-occurrence matrix. Metode penelitian ini terdiri dari : konversi data rgb ke grayscale, normalisasi citra, deteksi kualitas ikan cakalang menggunakan tekstur warna, ekstraksi fitur dan klasifikasi. Dalam penelitian ini digunakan data latih sebanyak 90 citra ikan cakalang yang merupakan data real. Data tersebut terdiri dari kualitas ikan cakalang di atas dan masing-masing terdiri dari yang Segar 45, dan Busuk 45. Data citra tersebut diolah menjadi citra grayscale yang kemudian dilakukan deteksi kualitas ikan cakalang menggunakan tekstur warna. Setelah didapat kualitas ikan cakalang kemudian dilakukan perhitungan ciri menggunakan gray level co-occurrence matrix. Algoritma yang digunakan untuk proses klasifikasi adalah algoritma k-nearest neighbor. Hasil akhir pengujian menunjukkan bahwa metode yang diusulkan telah mampu mendeteksi kualitas ikan cakalang menggunakan tekstur warna dengan hasil akurasi yang dihitung menggunakan confusion matrix, maka dapat ditarik suatu kesimpulan bahwa Pada percobaan model K-NN, dengan jumlah $K = 1$, dan Arah GLCM = 0° , dengan jarak = 1, maka mendapatkan nilai akurasi tertinggi Sebesar = 70%. Sedangkan pada percobaan model K-NN, dengan jumlah $K = 2$, dan Arah GLCM = 135° , dengan jarak = 2, maka mendapatkan nilai akurasi tertinggi sebesar = 60%. Pada percobaan model K-NN, dengan jumlah $K = 3$, dan Arah GLCM = 135° , dengan jarak = 1, Maka Mendapatkan Nilai Akurasi Sebesar = 60%. Pada percobaan model K-NN, dengan jumlah $K = 4$, dan Arah GLCM = 90° , dengan jarak = 4, Maka Mendapatkan Nilai Akurasi Sebesar = 80%. Pada percobaan dengan jumlah $K = 4$, dan Arah GLCM = 45° , dengan jarak = 4, maka mendapatkan nilai akurasi tertinggi Sebesar = 50%. Sedangkan pada percobaan dengan jumlah $K = 3$, dan Arah GLCM = 45° , dengan jarak = 3, maka mendapatkan nilai akurasi tertinggi sebesar = 70%. Dan dengan jumlah $K = 1$, dan Arah GLCM = 45° , dengan jarak = 1, Maka Mendapatkan Nilai Akurasi Sebesar = 50%. Pada percobaan model K-NN, dengan jumlah $K = 2$, dan Arah GLCM = 90° , dengan jarak = 2, Maka Mendapatkan Nilai Akurasi Sebesar = 40%. Dengan demikian penerapan metode gray level co-occurrence matrix dan k-nearest neighbor pada masalah deteksi kualitas ikan cakalang menggunakan tekstur warna perlu dikembangkan lagi hasil akurat.

Kata Kunci : Deteksi Kualitas, Gray Level Co-Occurrence Matrix, K-Nearest Neighbor

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamualaikum warahmatullahi wabarakatuh.

Alhamdulillah, penulis dapat menyelesaikan skripsi ini dengan judul “DETEKSI KUALITAS IKAN CAKALANG MENGGUNAKAN METODE *GLCM* DAN ALGORITMA *KNN*”, untuk memenuhi salah satu syarat penyusunan Skripsi Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penulis menyadari sepenuhnya bahwa skripsi 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 Muhamad Ichsan Gaffar, SE, M.Ak, Selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo.
2. Bapak Abdul Gaffar Latjokke, M.Si, Selaku Rektor Universitas Ichsan Gorontalo.
3. Ibu Zohrahayaty, S.Kom, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.
4. Bapak Sudirman S. Panna M.Kom, selaku Wakil Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Ichsan Gorontalo dan selaku Pembimbing I.
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 Melangi, M.Kom, selaku Wakil Dekan III Bidang Kemahasiswaan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.
7. Bapak Irfan A. Salihi, S.Kom, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.
8. Ibu Kartika Chandra Pelangi, M.Kom, selaku Pembimbing II

9. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan disiplin ilmu kepada penulis.
10. Kepada orang tua saya tercinta, atas segala kasih sayang, jerih payah dan doa restunya dalam membesarkan dan mendidik penulis.
11. Rekan-rekan seperjuangan yang telah telah banyak memberikan bantuan dan dukungan moril yang sangat besar kepada penulis.
12. Kepada semua pihak yang ikut membantu dalam penyelesaian proposal ini yang tak sempat penulis sebutkan satu-persatu.

Semoga Allah SWT melimpahkan balasan atas jasa-jasa mereka kepada kami. Penulis menyadari bahwa sepenuhnya bahwa apa yang telah dicapai masih jauh dari kata 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, 01 Juli 2020

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
PENGESAHAN SKRIPSI.....	ii
PERSETUJUAN SKRIPSI.....	iii
PERNYATAAN SKRIPSI.....	iv
ABSTRACT	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 Masalah	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. Kualitas Ikan Cakalang	6
2.2. 2. Jenis-Jenis Kualitas Ikan Cakalang.....	6
2.2.2. 1. Ikan Segar.....	6
2.2.2. 2. Ikan Busuk.....	7
2.2. 3. Pengolahan Citra Digital	7
2.2. 4. Jenis Citra Digital.....	8
2.2. 5. Proses Grayscale	9
2.2. 6. GLCM (<i>Gray Level Co-occurrence Matrix</i>).....	9

2.2. 7. KNN (<i>K-Nearst Neighbor</i>).....	13
2.2. 8. IMPLEMENTASI GLCM DAN KNN	14
2.2.8. 1. Analisis Perhitungan Data Masukan.....	14
2.2.8. 2. Ekstraksi Co-Occurrence Matrix	18
2.2.8. 3. Ekstraksi Ciri Fitur	22
2.2.8. 4. Analisis K-Nearest Neighbor.....	25
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. Pra Pengolahan	32
3.3.2. Ekstraksi Ciri.....	32
3.3.3. Data <i>Training</i>	32
3.3.4. <i>Training</i> Menggunakan KNN.....	32
3.3.5. Data <i>Testing</i>	33
3.3.6. Model.....	33
3.3.7. Hasil Klasifikasi	33
3.3.8. Evaluasi	33
BAB IV HASIL PENELITIAN.....	34
4. 1. Hasil Pengumpulan Data.....	34
4. 2. Hasil Pemodelan.....	39
4.2.1 Pra Pengolahan	39
4.2.2 Ekstraksi Fitur	42
4.2.3 Klasifikasi.....	45
4. 3. Evaluasi.....	53
4.3.1 Confusion Matrix.....	53
BAB V PEMBAHASAN	55
5. 1. Pembahasan Model	55
5. 2. Pembahasan Sistem.....	57
BAB VI PENUTUP	59

6. 1. Kesimpulan	59
6. 2. Saran.....	60
DAFTAR PUSTAKA	61
KODE PROGRAM.....	
1. Pembuatan Form	
2. Image Processing	
DAFTAR RIWAYAT HIDUP	

DAFTAR GAMBAR

Gambar 2.1 (Ikan Cakalang Segar).....	6
Gambar 2.2 (Ikan Cakalang Busuk).....	7
Gambar 2.3 (Tingkat <i>Grayscale</i>)	8
Gambar 2.4 (<i>Grayscale</i>).....	9
Gambar 2.5 (Piksel dengan berbagai sudut)	10
Gambar 2.6 (Contoh penentuan awal matriks GLCM).....	10
Gambar 2.7 (Matriks framework menjadi matriks simetris).....	11
Gambar 2.8 (Normalisasi matriks GLCM)	11
Gambar 2.9 (Penerapan KNN).....	13
Gambar 2.10 (Citra beras asli)	14
Gambar 2.11 (Citra Asli (Kiri) dan Citra Resize (Kanan)).....	15
Gambar 2.12 (Citra beras melalui Resize)	16
Gambar 2.13 (Nilai RGB beras 3x3).....	16
Gambar 2.14 (Matrix Greyscale Citra Beras 3x3)	17
Gambar 2.15 (Matriks Greyscale citra Beras range 0-15)	18
Gambar 2.16 (Citra Greyscale range 0-15)	18
Gambar 3.1 (Desain Penelitian)	30
Gambar 3.2 (Pemodelan)	31
Gambar 5.1 (Tampilan Menu Utama).....	57

DAFTAR TABEL

Tabel 2.1 (Tinjauan Studi)	5
Tabel 2.2 (Framework rentang 0-15)	18
Tabel 2.3 (Co-Occurrence Matrix).....	19
Tabel 2.4 (Transpose Matriks)	20
Tabel 2.5 (Simetris Matriks)	20
Tabel 2.6 (Normalisasi Matriks)	21
Tabel 2.7 (Matriks Kontras).....	22
Tabel 2.8 (Matriks Homogen).....	23
Tabel 2.9 (Matriks Entropy).....	24
Tabel 2.10 (Matriks Energy)	25
Tabel 2.11 (Nilai Hasi Fitur Ekstraksi 0^0)	25
Tabel 2.12 (Data Fitur Ekstraksi Ciri).....	26
Tabel 2.13 (Data Fitur Ekstraksi Baru)	26
Tabel 2.14 (Hasil Pengurutan Fitur).....	27
Tabel 2.15 (Kelas Fitur Terdekat).....	28
Tabel 2.16 (Hasil Akhir)	28
Tabel 4.1 (Dataset Ikan Cakalang).....	34
Tabel 4.2 (Mengkonversi RGB ke Grayscale).....	36
Tabel 4.3 (Citra Dengan Image 4x4).....	36
Tabel 4.4 (Perhitungan Dengan Langkah Dan Rumus)	38
Tabel 4.5 (Citra 4x4 Asli)	38
Tabel 4.6 (Data Citra 4x4 Setelah Dilakukan Histogram Equalisation)	38
Tabel 4.7 (Proses Perubahan Citra Asli Menjadi <i>Grayscale</i>).....	39
Tabel 4.8 (Proses Normalisasi Citra)	40
Tabel 4.9 (Hasil Segmentasi Otsu).....	41
Tabel 4.10 (Nilai dari suatu citra)	42
Tabel 4.11 (Matrix GLCM).....	43
Tabel 4.12 (Matrix Ternormalisasi)	43
Tabel 4.13 (Hasil Perhitungan Data Training).....	46
Tabel 4.14 (Hasil Perhitungan Data Testing).....	48

Tabel 4.15 (Perbandingan Nilai)	51
Tabel 4.16 (Hasil Pengurutan dan Nilai Terkecil berdasarkan perhitungan jarak dari Masing-masing Data Training ke Data Testing).....	52
Tabel 4.17 (Hasil Jumlah K yang Sesuai Data)	52
Tabel 4.18 (Hasil Data Testing)	53
Tabel 4.19 (Hasil Data Uji).....	54

DAFTAR LAMPIRAN

KODE PROGRAM.....	
DAFTAR RIWAYAT HIDUP	
SURAT KETERANGAN PENELITIAN.....	
SURAT BEBAS PLAGIASI.....	
HASIL TURNITIN	

BAB I

PENDAHULUAN

1. 1. Latar Belakang Masalah

Dunia perikanan di lautan Indonesia tepatnya di Gorontalo merupakan salah satu wilayah yang banyak menggunakan kecanggihan teknologi, walaupun sebagian masih ada yang menggunakan penilaian secara manusiawi dalam bentuk menentukan mutu kesegaran ikan cakalang. Secara umum ikan diperjual belikan dalam keadaan yang sudah mati atau juga dalam keadaan masih hidup [1]. Ikan cakalang adalah jenis ikan laut yang sering dikonsumsi oleh masyarakat lokal. Ikan cakalang memiliki kandungan protein tinggi sehingga ikan cakalang tergolong sumberdaya perikanan yang paling banyak di minati oleh masyarakat indonesia karena ikan ini mengandung gizi dan protein yang sangat baik bagi tubuh manusia [2].

Menurut Balai Pengujian, Penerapan Mutu (UPTD-BPPMDPP) Dinas Kelautan dan Perikanan Provinsi Gorontalo, Dalam rangka memberikan jaminan mutu dan keamanan pangan komoditas ikan segar yang akan dipasarkan di dalam dan luar negeri, maka ikan yang dipasarkan harus memenuhi semua ketentuan yang terdapat dalam Standar Nasional Indonesia Nomor 01-2729.1 Tahun 2006 tentang Spesifikasi Ikan Segar. Mengenai masalah yang sering ditemukan di Gorontalo penurunan kualitas ikan sejak diangkat dari air sampai ke tangan konsumen, serta mudahnya terjadi pembusukan terhadap ikan di karenakan adanya bakteri atau enzim di dalamnya yang menyebabkan ikan tersebut cepat membusuk kalau tidak di tangani dengan cepat. Karena ikan cakalang tersebut sejak di angkat dari air membutuhkan penanganan yang khusus disebabkan dengan kulitnya yang begitu halus dan mudah rusak.

Dalam menjaga kualitas produksi ikan cakalang maka diperlukan penanganan yang cepat dan cermat dalam upaya mempertahankan mutunya sejak ikan tersebut diangkat dari air. Salah satu metode yang digunakan untuk menjaga tekstur ikan cakalang yaitu menggunakan metode pendinginan agar kualitas ikan

cakalang tersebut tetap terjaga. Umumnya penanganan ikan segar di Indonesia, terutama yang dilakukan oleh para nelayan masih sangat memprihatinkan. Mempertahankan kesegaran ikan hasil tangkapan sangat penting demi mendapatkan mutu ikan yang baik sehingga memiliki nilai jual yang tinggi. [3], Ikan dikenal sebagai suatu yang mempunyai nilai gizi tinggi namun mudah membusuk karena mengandung kadar protein yang tinggi dengan kandungan asam amino bebas yang digunakan untuk metabolisme mikroorganisme, produksi amonia, biogenik amin, asam organik, keton dan komponen sulfur. karena konsumen menuntut penampilan, bau, rasa dan tekstur yang baik [4]. Penurunan mutu ikan dapat terjadi mulai dari saat penangkapan dan terus berlangsung hingga ke tangan konsumen akhir [5].

Citra ikan yang mengalami penurunan kualitas dapat diamati berdasarkan bentuk, pola tubuh ikan, beserta perubahan ciri-cirinya. Pengidentifikasian kualitas ikan menggunakan pendekatan pengolahan citra digital, dengan demikian pengenalan terhadap perubahan warna citra ikan akan lebih akurat [6]. Maka dari sebab itu dibutuhkan sebuah sistem yang bisa mendeteksi perbedaan kualitas kesegaran pada ikan cakalang agar dapat memudahkan industri dalam menjaga kualitas ikan cakalang segar yang bermutu dan dapat memuaskan konsumen. Teknologi yang digunakan adalah pengolahan citra atau image processing yang mempelajari teknik-teknik pengolahan citra atau gambar.

Metode GLCM (Gray-Level Cooccurrence Matrix), yang digunakan untuk menganalisa Citra digital. Teknik ini sangat kental dengan matematika, seiring dengan pesatnya komputasi dengan bantuan komputer menjadikan penemuan ini lebih bermanfaat dan banyak digunakan. Sehingga memudahkan para peneliti untuk mendapatkan nilai pada fitur ekstraksi dan mendapatkan tingkat akurasi yang lebih tinggi [7]. Berdasarkan penelitian di atas [8] membuktikan bahwa dari hasil penelitian mereka metode GLCM adalah metode yang paling efektif digunakan dan fitur ekstraksi yang paling baik dari fitur ekstraksi lain.

Dari para peneliti sebelumnya pernah menggunakan metode K-Nearest Neighbor dalam mengklasifikasi suatu data atau objek adalah suatu teknik yang sangat efektif dan efisien. Teknik K-Nearest Neighbor ini sangat sederhana dan

mudah diimplementasikan dan bisa menghasilkan tingkat akurasi yang lebih tepat, serta bisa melakukan training data dalam jumlah yang lebih besar [9].

Adapun peneliti lain pernah meneliti dengan judul penelitian Identifikasi Tumbuhan Obat Herbal Berdasarkan Citra Daun Menggunakan Algoritma Gray Level Co-occurrence Matrix dan K-Nearest Neighbor berdasarkan jarak terdekat antara citra uji dan citra latih, Algoritma tersebut dapat digunakan untuk mengidentifikasi tumbuhan obat herbal berdasarkan citra daun dengan akurasi rata-rata sebesar 83,33% [10].

Peneliti dari STMIK Lombok pernah melakukan penelitian dengan judul Sistem pengenalan kualitas Ikan Gurame dengan Wavelet, PCA, Histogram HSV dan Klasifikasi KNN, mendapatkan tingkat akurasi secara keseluruhan sebesar 89,5% [11].

Adapun peneliti sebelumnya pernah melakukan penelitian dengan judul Penerapan Metode K-Nearest Neighbor dalam Identifikasi Kesegaran Ikan Mujair menggunakan Metode GLCM dan KNN mendapatkan tingkat akurasi sebesar 90% [12].

Berdasarkan paparan di atas, penulis akan menggunakan metode GLCM dan KNN untuk mengetahui kualitas dari ikan cakalang tersebut dengan judul “***Deteksi Kualitas Ikan Cakalang Menggunakan Metode GLCM dan Algoritma KNN***”.

1. 2. Identifikasi Masalah

Berdasarkan uraian latar belakang di atas maka terdapat masalah yaitu kualitas kesegaran ikan cakalang ketika di angkat dari air mudah terjadi pembusukan pada ikan dan cara yang di gunakan masih manual dengan laboratorium.

1. 3. Rumusan Masalah

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

1. 4. Tujuan Penelitian

1. Untuk mengetahui hasil dari metode GLCM dan algoritma KNN yang akan mengklasifikasikan kualitas ikan cakalang berdasarkan citra atau gambar.
2. Untuk memperoleh hasil klasifikasi kualitas ikan cakalang dengan metode GLCM dan algoritma KNN

1. 5. Manfaat Penelitian

1. 5. 1. Manfaat Teoritis

Manfaat Teoritis dalam penelitian ini adalah :

1. Di harapkan penelitian ini bisa menambah ilmu pengetahuan di bidang sistem komputer khususnya dalam Computer Vision untuk bisa mendeteksi perbedaan kualitas ikan cakalang.
2. Bisa membantu industri dalam memproduksi kualitas kesegaran ikan cakalang.

1. 5. 2. Manfaat Praktis

Manfaat penelitian metode GLCM untuk mendeteksi kualitas kesegaran ikan cakalang dengan menggunakan metode KNN ini dapat digunakan sebagai pengambilan keputusan bahwa ikan cakalang tersebut masih segar atau tidak berdasarkan citra atau gambar.

BAB II

LANDASAN TEORI

2. 1. Tinjauan Studi

Tabel 2.1: Tinjauan Studi

NO	PENELITI	JUDUL	TAHUN	METODE	HASIL
1.	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	83,33%
2.	Fitri Astutik	Sistem Pengenalan Ikan Gurame dengan Wavelet, PCA, Histogram HSV dan KNN	2013	Wavelet haar, Principle Component Analysis (PCA), Histogram HSV, dan Klasifikasi KNN	89,5%
3.	Nikolaus Wesli Roberto Situmorang	Penerapan Metode K- Neighbor dalam Identifikasi Kesegaran Ikan Mujair menggunakan Metode GLCM dan KNN	2019	Pengolahan Citra Digital menggunakan Metode GLCM dan Klasifikasi KNN	90%

2. 2. Tinjauan Pustaka

2.2. 1. Kualitas Ikan Cakalang

Kualitas Ikan Cakalang adalah ikan ini memiliki banyak kandungan protein yang tinggi dan baik bagi tubuh manusia, ikan cakalang biasa di konsumsi oleh masyarakat di Indonesia. Ikan yang masih segar ialah ikan yang masih mempunyai sifat yang sama dengan ikan yang masih hidup, baik dari rupanya, bau, rasa, dan teksturnya. Adapun penurunan kualitas ikan cakalang tersebut bisa dilihat dari perubahan warna kulit ikan, mata, insang dan tekstur daging ikan cakalang. Perubahan ini disebabkan oleh adanya aktivitas bakteri di dalamnya karena sejak ikan itu mati, ikan cakalang tersebut mudah membusuk dan ikan yang sudah busuk tersebut tidak layak di perdagangan apalagi sampai di konsumsi oleh manusia [2].

2.2. 2. Jenis-Jenis Kualitas Ikan Cakalang

2.2.2.1 Ikan Segar

Ikan Segar adalah Ikan yang masih mempunyai sifat sama dengan ikan yang masih hidup, baik rupa, bau, rasa dan teksturnya. Ikan tersebut masih mempunyai kenampakan yang cerah, tekstur daging ikan yang masih keras bila di tekan tidak menimbulkan bekas dan yang terakhir pada bagian matanya yaitu ikan yang masih segar tersebut matanya masih cerah, segar, jernih dan cembung menonjol [1].



Gambar 2.1: Ikan Cakalang Segar

2.2.2.2 Ikan Busuk

Ikan Busuk adalah Penurunan kualitas ikan dapat dilihat dari perubahan warna kulit ikan, mata, insang dan tekstur daging ikan. Perubahan perubahan tersebut disebabkan oleh adanya aktivitas enzim, kimiawi, dan bakteri didalamnya sehingga menyebabkan ikan tersebut tidak layak untuk diperdagangkan apalagi di konsumsi oleh manusia. Penurunan kualitas ikan tersebut bisa di kenali dari tekstur daging yang rusak, bau yang menyegat serta dapat ditentukan dari mata ikan tersebut berbentuk cekung dan keruh [1].



Gambar 2.2 : Ikan Cakalang Busuk

2.2. 3. 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. 4. 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). Masing-masing komponen menggunakan delapan bit yang nilainya berkisar antara 0 sampai dengan 255 [16].

- b. *Grayscale*

Grayscale adalah suatu 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 2.3.



Gambar 2.3 : Tingkat *Grayscale* [18]

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

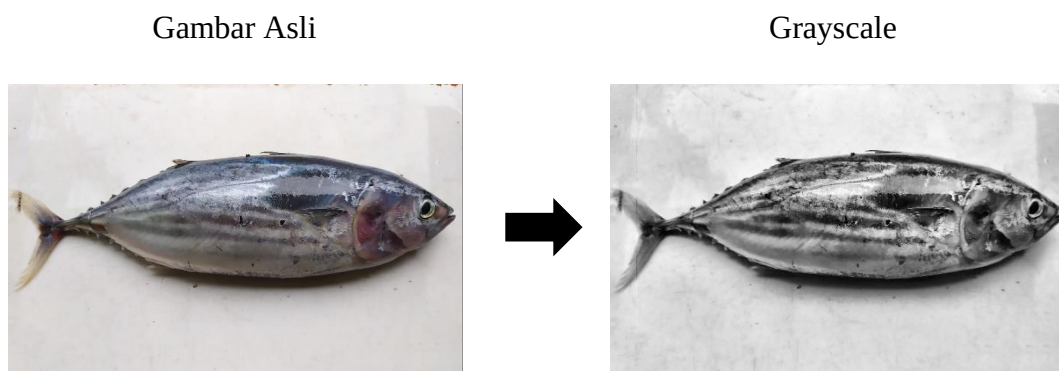
$$s = \frac{r + g + b}{3} \quad (1)$$

a. *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. 5. 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].

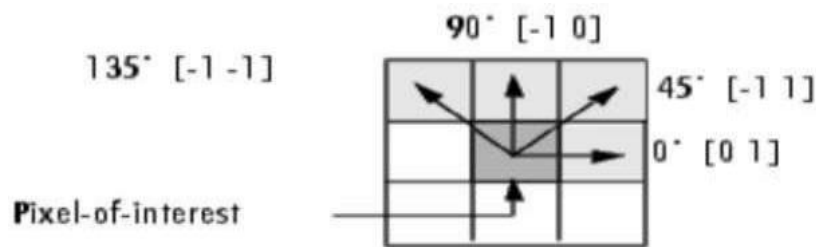


Gambar 2.4 : Grayscale [19]

2.2. 4. GLCM (Gray Level Co-Occurrence Matrix)

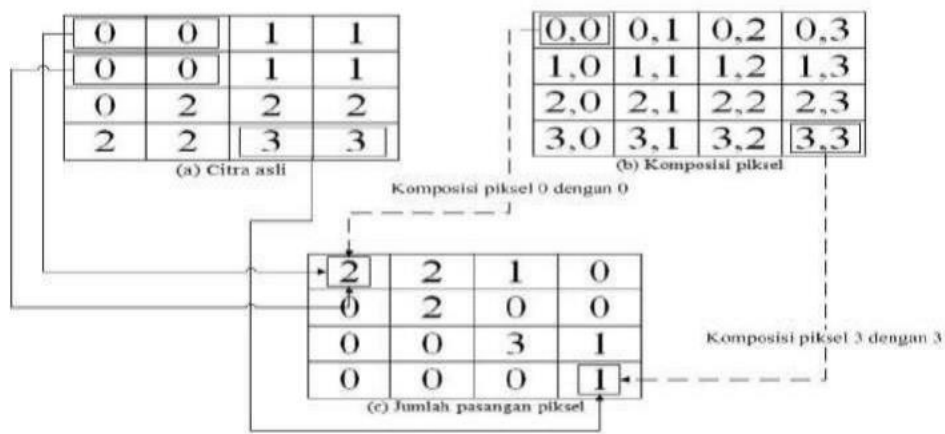
Gray Level Co-occurrence Matrix (GLCM) adalah suatu metode yang digunakan untuk analisis tekstur/ekstraksi ciri. GLCM merupakan suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra [20]. Gray Level Co-Occurance Matrices (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 2.5 : 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 2.6 : Contoh penentuan awal matriks GLCM [22].

Matriks pada Gambar 2.6 dinamakan matrix framework. Matriks ini kemudian diolah menjadi matriks dimetris dengan cara menambahkan hasil transposnya [12]. Seperti yang terlihat pada Gambar 2.7.

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

Tranpos
GLCM sebelum dinormalisasi

Gambar 2.7 : 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 2.8.

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

Gambar 2.8 : 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.

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

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 :

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

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 P_{ij} yang tinggi, oleh karena itu jumlah dari pangkat P_{ij} akan tinggi.

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

2.2. 7. KNN (*K-Nearest Neighbor*)

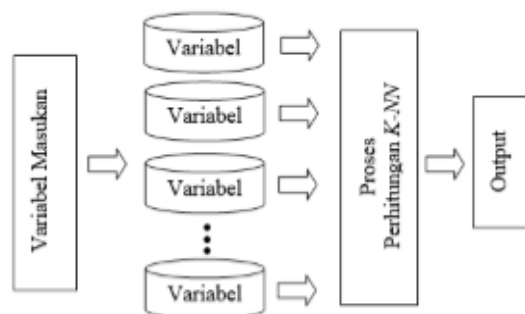
Algoritma K-Nearest Neighbor (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. 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 2.9 : 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. 8. 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].

2.2.8.1 Analisis Perhitungan Data Masukan

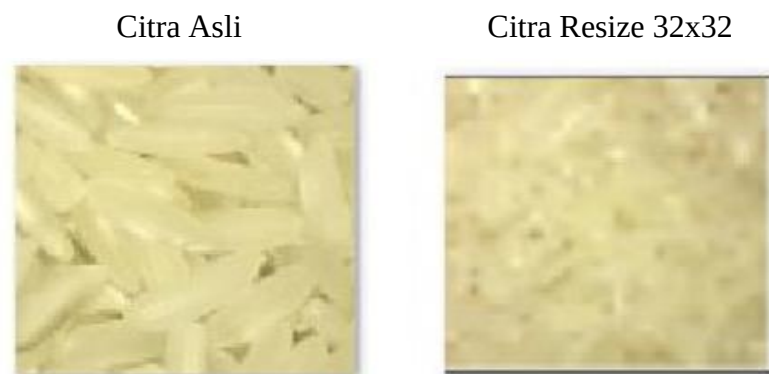
Data masukan berupa citra beras dengan berbagai kualitas dan akan melalui beberapa tahapan proses pengolahan citra, diantaranya adalah proses resizing, greyscaling, kuantisasi citra grayscale dan ekstraksi ciri. Berikut data masukan :



Gambar 2.10 : 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 2.11 : Citra Asli (Kiri) dan Citra Resize (Kanan) [24]

2. Greyscale

Setelah melalui tahap resize, maka tahap selanjutnya adalah mengubah citra mode 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 :

$$X = 0.21 * R + 0.72 * G + 0.07 * B \quad (7)$$

Dimana

X = Nilai Greyscale

R = Nilai Red

G = Nilai Green

B = Nilai Blue

Analisis perhitungan :

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



Gambar 2.12 : 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 2.13 : 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 2.14 : 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 waktu pemrosesan 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 2.15 : Matriks Greyscale citra Beras range 0-15 [24]

2.2.8.2 Ekstraksi Co-occurence Matrix

Metode Matrix Co-occurrence merupakan metode untuk ekstraksi ciri, dimana nilai ekstraksi ciri yang akan dicari adalah nilai *Contras*, *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 2.16 : 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

2.2.8.3 Ekstraksi Ciri Fitur

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

1. Perhitungan Kontras

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

Dimana :

i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatanp (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	0,028226	0,008056	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2. 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	0,000201	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

3. Perhitungan Entropy

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

Dimana :

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

Contoh perhitungan :

$$-(P(i,j) \log(P(i,j)))$$

$$= -(0,002014 \cdot \log(0,002014))$$

$$= 0.00542962$$

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

4. Perhitungan Energy

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

Dimana :

i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatan p (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.

2.2.8.4 Analisis K-Nearest Neighbor

K-Nearest Neighbor merupakan sebuah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasi 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

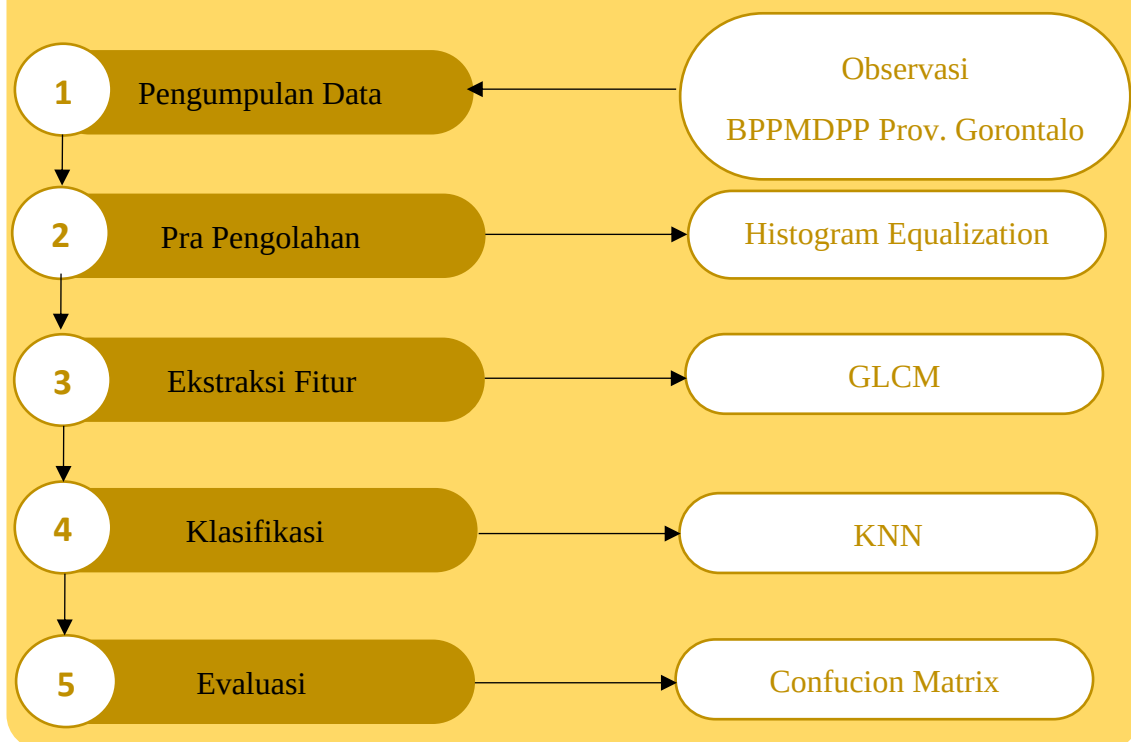
Jadi kesimpulan dari contoh studi kasus di atas adalah hasil dari kualitas yang di cari adalah KW3 karena perbandingan dengan kelas baru yang jumlahnya hanya sedikit di banding dengan kualitas KW3, jadi hasil dari output pencarian kualitas di atas adalah mayoritas Kualitas KW3.

2. 3. Kerangka Pikir

MASALAH

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

PEMODELAN/ABSTRAK



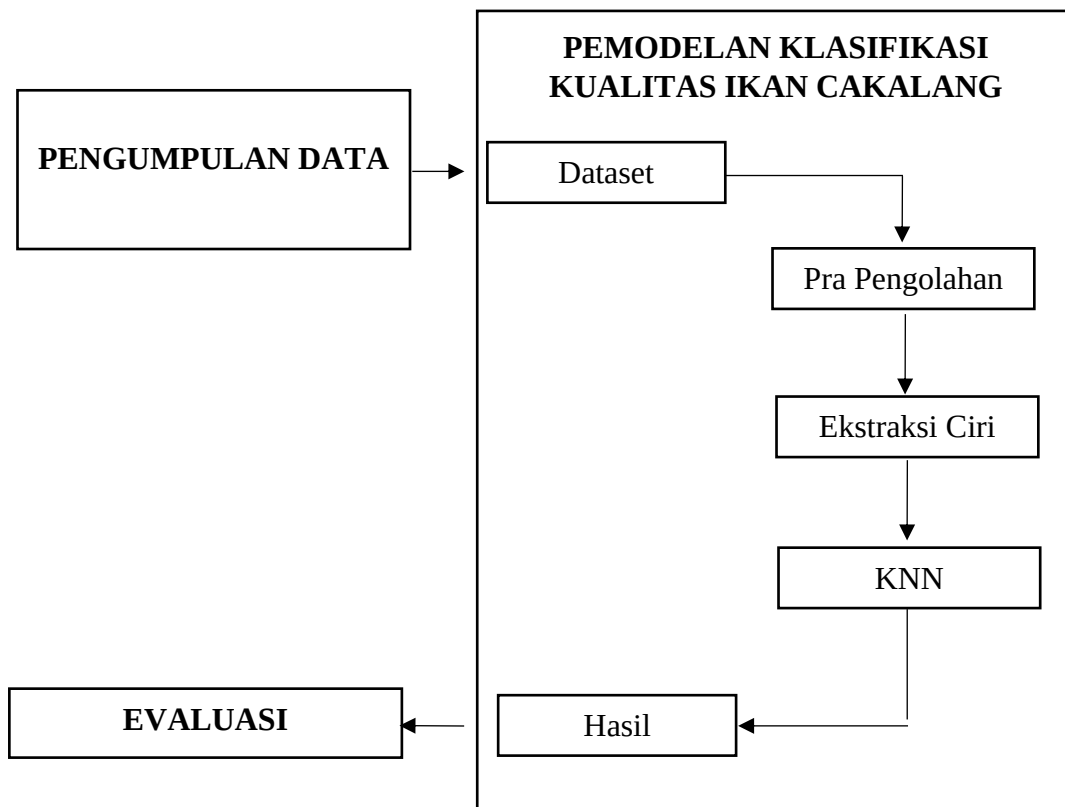
BAB III

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 *eksperimental*.

Subjek penelitian ini adalah Klasifikasi pada objek citra pada ikan cakalang. Penelitian ini dimulai dari September 2019 sampai Desember 2019.

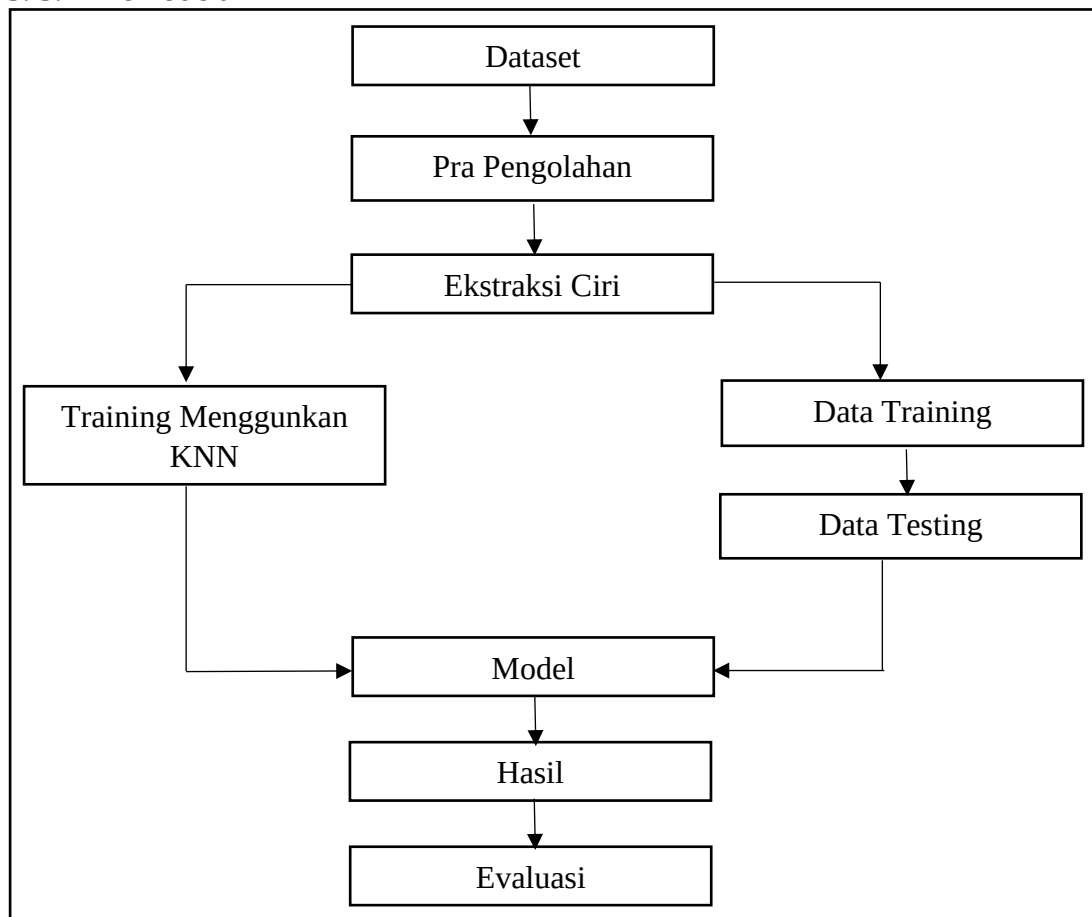


Gambar 3.1: Desain Penelitian

3. 2. Pengumpulan Data

Data yang dikumpulkan dari penelitian adalah data hasil observasi yang sudah peneliti lakukan dan juga dari Menurut Balai Pengujian, Penerapan Mutu (UPTD-BPPMDPP) dan Dinas Kelautan dan Perikanan Provinsi Gorontalo kualitas pada ikan cakalang yang sering ditemukan adalah penurunan kualitas ikan sejak diangkat dari air sampai ke tangan konsumen, serta mudahnya terjadi pembusukan terhadap ikan di karenakan adanya bakteri atau enzim di dalamnya yang menyebabkan ikan tersebut cepat membusuk. Citra yang dikumpulkan dibagi menjadi dua dataset yaitu: data latih dan data uji. Data latih yang digunakan sebagai pembandingan terhadap data uji pada proses pendeteksian kualitas pada ikan cakalang. Sedangkan data uji adalah data digunakan untuk mengetahui hasil deteksi jenis kualitas pada ikan cakalang yang didapatkan melalui proses deteksi.

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 cakalang 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 cakalang 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 cakalang.

3.3. 4. Training Menggunakan KNN

Training menggunakan K-NN menjadikan data training yang berupa angka atau parameter hasil dari citra kualitas ikan cakalang 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 cakalang yang telah didapatkan untuk dijadikan prediksi dari test.

3.3. 5. 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 cakalang 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 cakalang.

3.3. 6. Model

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

3.3. 7. 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. 8. Evaluasi

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

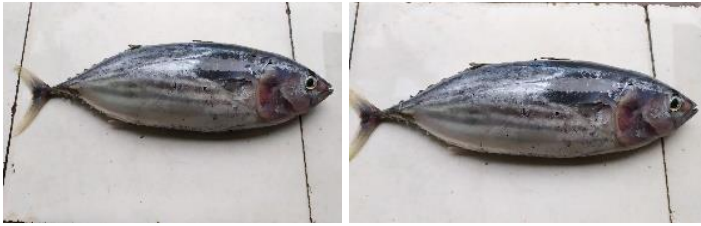
BAB IV

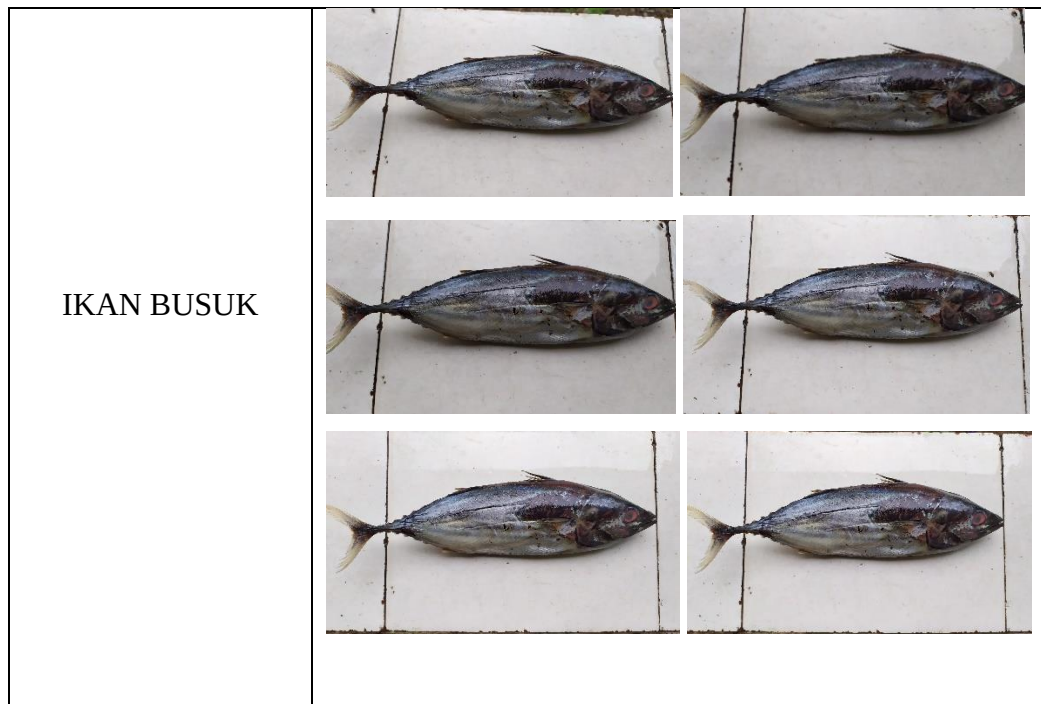
HASIL PENELITIAN

4. 1. Hasil Pengumpulan Data

Seperti yang telah diuraikan pada tahapan metode penelitian bahwa dataset gambar yang digunakan pada penelitian ini ada berupa data real, kami telah melakukan proses pengumpulan data tersebut dan berhasil di dapat, Data terdiri 18 citra dan 3 objek dengan format file gambar adalah jpg. Berikut sampel gambar yang berhasil dilakukan :

Tabel 4.1 Dataset Ikan Cakalang

IKAN CAKALANG	PENGUMPULAN DATASET
IKAN SEGAR	
	
	



Untuk melakukan proses analisis pada penelitian ini adalah dengan melakukan pemilihan data gambar yang dijadikan sebagai data training dan data testing, sehingganya dari pekerjaan yang kami lakukan pada penelitian ini kami menentukan dari data yang telah berhasil kami dapatkan ditentukan jumlah data training adalah sebanyak 18 data gambar dan data testing adalah sebanyak 3 data gambar.

Tabel 4.2 Mengkonversi RGB ke Grayscale

Daftar Gambar	Nilai R, G, B	Hasil
	$196R + 187G + 182B / 3$ $197R + 188G + 183B / 3$ $196R + 187G + 182B / 3$	188,3 189,3 188,3

HISTOGRAM EQUALIZATION

Histogram Equalization adalah suatu proses perataan histogram, dimana distribusi nilai derajat keabuan pada suatu citra dibuat rata. Pada dasarnya metode ini akan digunakan untuk memperlebar range tingkat keabuan, sehingga akan meningkatkan kekontrasan citra. Dari hasil penurunan rumus perataan Histogram secara matematis didapat tranformasi sebagai berikut:

$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j)$$

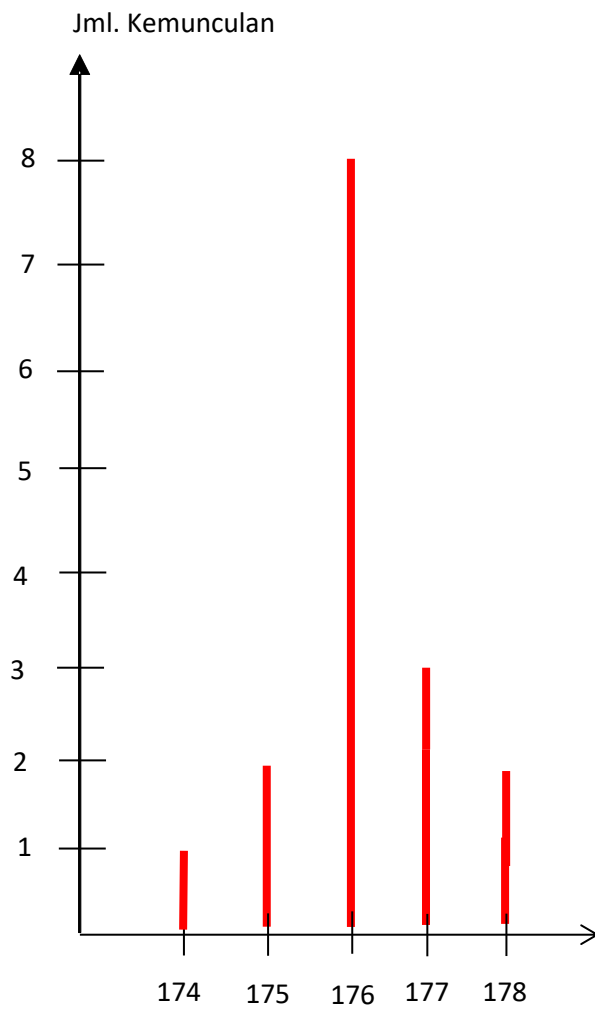
$$= \sum_{j=0}^k \frac{n_j}{n}$$

Sebagai Contoh didapat sebuah citra dengan image 4x4

Tabel 4.3 Citra Dengan Image 4x4

176	176	176	175
177	177	176	177
175	176	176	178
176	176	174	178

Dengan Histogram sebagai berikut :



Setelah didapat histogram asli dari citra diatas, maka dilakukan perhitungan dengan langkah dan rumus pada tabel dibawah :

Tabel 4.4 Perhitungan Dengan Langkah Dan Rumus

Derajat Keabuan (j)	174	175	176	177	178
Kemunculan	1	2	8	3	2
$s = \sum_{j=0}^k \frac{n_j}{n}$	1/16	2/16	8/16	3/16	2/16

Hasil dari perhitungan Perataan Histogram diatas telah menunjukan sejumlah data citra baru seperti pada tabel diatas, dan disimpulkan seperti berikut :

Tabel 4.5 Citra 4x4 Asli

176	176	176	175
177	177	176	177
175	176	176	178
176	176	174	178



Tabel 4.6 Data Citra 4x4 Setelah Dilakukan Histogram Equalisation

0,5	0,5	0,5	0,125
0,1875	0,1875	0,5	0,1875
0,125	0,5	0,5	0,125
0,5	0,5	0,0625	0,125

4. 2. Hasil Pemodelan

4. 2. 1. Pra Pengolahan

Pra-pengolahan citra (image pre-processing), yaitu proses paling awal dalam pengolahan citra sebelum proses utama dilakukan. Pada tahap ini citra ikan cakalang yang sudah ada dikonversi agar diperoleh data citra ikan cakalang yang sesuai kebutuhan. Tahap ini berfungsi untuk menormalisasi citra ikan cakalang dari permasalahan luminasi yang terlalu gelap atau terlalu terang sehingga dapat meningkatkan performansi dari sistem pengenalan ikan cakalang. Pra-pengolahan dalam penelitian ini dilakukan dengan 4 tahap, yaitu:

1. Pengubahan citra warna ke *grayscale*

Pra-pengolahan pertama yang akan dilakukan adalah merubah citra *training* atau citra *testing* yang awalnya berbentuk citra dari RGB (red, green, blue) menjadi citra bentuk 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. Berikut gambar proses perubahan citra warna ke grayscale

Tabel 4.7 Proses Perubahan Citra Asli Menjadi *Grayscale*

Perbedaan	<i>Images Original</i>	<i>Images Grayscale</i>
IKAN CAKALANG SEGAR		

IKAN CAKALANG BUSUK		
------------------------------------	---	---

2. Normalisasi citra

Setelah citra asli di konversi ke citra abu-abu maka pra-pengolahan selanjutnya adalah normalisasi citra dengan histogram ekualisasi. Histogram ekualisasi adalah sebuah proses yang mengubah distribusi nilai derajat keabuan pada sebuah citra sehingga menjadi seragam. Tujuan dari histogram equalization adalah untuk memperoleh penyebaran histogram yang merata sehingga setiap derajat keabuan memiliki jumlah piksel yang relatif sama. Berikut gambar normalisasi citra abu-abu dengan histogram ekualisasi.

Tabel 4.8 Proses Normalisasi Citra

Perbedaan	Citra <i>Grayscale</i>	Citra <i>Histogram</i>
IKAN CAKALANG SEGAR		



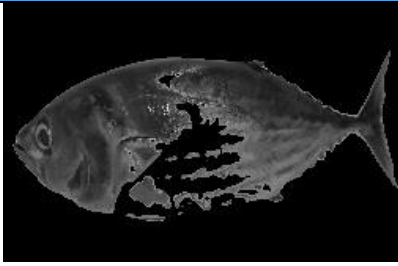
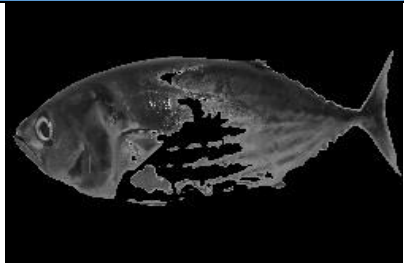
3. Segmentasi Citra

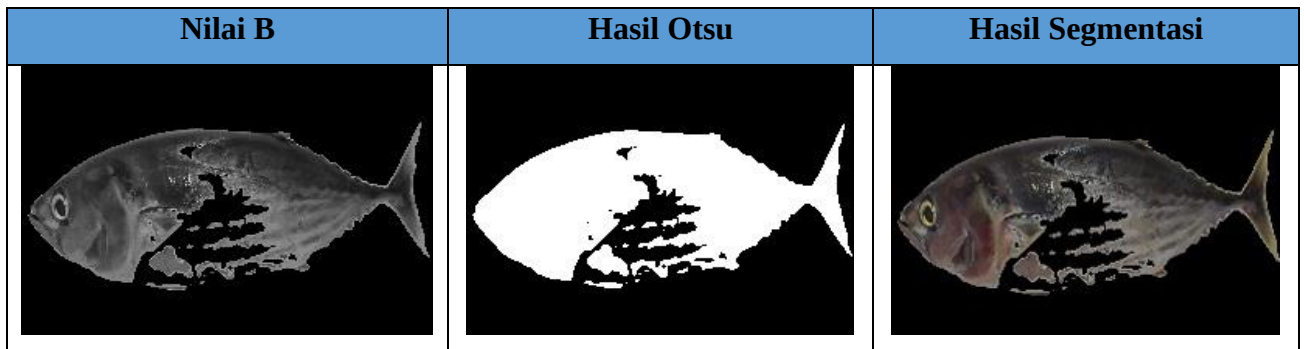
Segmentasi citra merupakan bagian dari proses pengolahan citra. Segmentasi citra (*image segmentation*) mempunyai arti membagi suatu citra menjadi wilayah-wilayah yang homogen berdasarkan kriteria keserupaan tertentu antara tingkat keabuan suatu piksel dengan tingkat keabuan piksel-piksel tetangganya, kemudian hasil dari proses segmentasi ini akan digunakan untuk proses lebih lanjut [8].

4. Metode Otsu *Thresholding*

Metode Otsu merupakan salah satu metode untuk segmentasi citra digital dengan menggunakan nilai ambang secara otomatis, yakni mengubah citra digital warna abu-abu menjadi hitam putih berdasarkan perbandingan nilai ambang dengan nilai warna piksel citra digital. Metode Otsu *Thresholding* diperkenalkan pertama kali oleh Nobuyuki Otsu, dalam jurnal ilmiahnya yang berjudul “*A Threshold Selection Method from Grayscale Histogram*” pada tahun 1979 [9].

Tabel 4.9 Hasil Segmentasi Otsu

Citra Asli	Nilai R	Nilai G
		



4. 2. 2. Ekstraksi Fitur

Fitur ekstraksi Grey Level Co-occurrence Matrix (GLCM) merupakan matrix yang menggambarkan frekuensi munculnya pasangan dua pixsel dengan intensitas tertentu dalam jarak d dan orientasi arah sudut 0° , 90° , 135° , atau 45° tertentu dalam citra.

Contoh: Jarak 1 dan orientasi arah sudut 0°

Nilai dari Gambar 4.10 berupa matrix 10×10 Yang sebenarnya matrix 104×104 dan di ambil dari 1 contoh gambar 1 dataset ikan busuk dari cakalang yang disederhanakan unruk memudahkan perhitungan :

Tabel 4.10 Nilai dari suatu citra

i/j	0	1	2	3	4	5	6	7	8	9
0	227	224	229	221	226	224	227	228	218	219
1	229	224	225	218	221	223	226	228	218	219
2	229	228	230	229	228	230	229	228	230	229
3	233	231	233	233	229	232	231	232	232	234
4	232	232	235	236	232	234	233	233	231	235
5	232	231	232	232	231	232	232	232	234	235
6	232	231	232	232	232	233	232	232	232	233
7	231	231	232	232	232	233	233	234	234	233
8	229	231	231	232	232	232	233	234	233	233
9	229	231	230	232	234	234	234	233	234	233

Pada tabel berikut ini nilai yang sudah di normalisasi dari tabel sebelumnya, yaitu nilai yang di cari kesamaan berdasarkan jarak 1 dan arah 0^0 .

Tabel 4.11 Matrix GLCM

i/j	226	228	229	231	232	234
226	0	0	0	0	0	0
228	1	0	3	0	0	0
229	0	3	0	2	1	0
231	0	0	2	1	6	0
232	0	0	0	6	9	2
234	0	0	0	0	4	2

jumlah total nilai GLCM = 42.

Selanjutnya pada tabel 4.12 nilai dalam pixel sebelumnya di jumlahkan dengan jumlah total dalam pixel sebelumnya.

Tabel 4.12 Matrix Ternormalisasi

i/j	226	228	229	231	232	234
226	0	0	0	0	0	0
228	0.024	0	0.071	0	0	0
229	0	0.071	0	0.048	0.024	0
231	0	0	0.048	0.024	0.143	0
232	0	0	0	0.143	0.214	0.048
234	0	0	0	0	0.095	0.048

Setelah itu nilai kontras, homogeneity, entropy dan energy di hitung secara manual seperti di bawah ini:

a. Menghitung nilai *Contras*

$$\text{Contras} = ((228-226)^2(0.024) + (228-229)^2(0.071) + (229-228)^2(0.071) + (229-231)^2(0.048) + (229-232)^2(0.024) + (231-229)^2(0.048) + (231-231)^2(0.024) +$$

$$(231-232)^2(0.143) + (232-231)^2(0.143) + (232-232)^2(0.214) + (232-234)^2(0.048) \\ + (234-232)^2(0.095) + (234-234)^2(0.048))$$

$$\text{Contras} = 4*0.024 + 1*0.071 + 1*0.71 + 4*0.048 + 9*0.024 + 4*0.048 + 0*0.024 \\ + 1*0.143 + 1*0.143 + 0 + 4*0.048 + 4*0.095 + 0$$

$$\text{Contras} = 0.096 + 0.071 + 0,071 + 0.192 + 0.216 + 0.192 + 0 + 0.143 + 0.143 + 0 \\ + 0.192 + 0.38 + 0$$

$$\text{Contras} = \mathbf{1.696}$$

b. Menghitung nilai *Energy*

$$\text{Energy} = ((0.024)^2) + ((0.071)^2) + ((0.071)^2) + ((0.048)^2) + ((0.024)^2) + ((0.048)^2) \\ + ((0.024)^2) + ((0.143)^2) + ((0.143)^2) + ((0.214)^2) + ((0.048)^2) + ((0.095)^2) + \\ ((0.048)^2)$$

$$\text{Energy} = 0.024*0.024 + 0.071*0.071 + 0.071*0.071 + 0.048*0.048 + 0.024*0.024 \\ + 0.048*0.048 + 0.024*0.024 + 0.143*0.143 + 0.143*0.143 + 0.214*0.214 + \\ 0.048*0.048 + 0.095*0.095 + 0.048*0.048$$

$$\text{Energy} = 0.00576 + 0.005041 + 0.005041 + 0.002304 + 0.000576 + 0.002304 + \\ 0.000576 + 0.020449 + 0.020449 + 0.045796 + 0.002304 + 0.009025 + 0.002304$$

$$\text{Energy} = \mathbf{0.121979}$$

c. Menghitung nilai *Homogeneity*

$$\text{Homogeneity} = ((0.024) / 1 + (228-226)) + ((0.071) / 1 + (228-229)) + ((0.071) / 1 \\ + (229-228)) + ((0.048) / 1 + (229-231)) + ((0.024) / 1 + (229-232)) + ((0.048) / 1 \\ + (231-229)) + ((0.024) / 1 + (231-231)) + ((0.143) / 1 + (231-232)) + ((0.143) / 1 \\ + (232-231)) + ((0,214) / 1 + (232-232)) + ((0.048) / 1 + (232-234)) + ((0.095) / 1 \\ + (234-232)) + ((0.048) / 1 + (234-234))$$

$$\text{Homogeneity} = ((0.024 + 2)) + ((0.071 + (-1))) + ((0.071 + 1)) + ((0.048 + (-2))) + \\ ((0.024 + (-3))) + ((0.048 + 2)) + ((0.024 + 0)) + ((0.143 + (-1))) + ((0.143 + 1)) + \\ ((0.214 + 0)) + ((0.048 + (-2))) + ((0.095 + 2)) + ((0.048 + 0))$$

$$\text{Homogeneity} = 2.024 + (-0.929) + 1.071 + (-1.952) + (-2.976) + 2.048 + 0.024 + (-0.857) + 1.143 + 0.214 + (-1.952) + 2.095 + 0.048$$

$$\text{Homogeneity} = \mathbf{0.001}$$

d. Menghitung nilai *Entropy*

$$\begin{aligned} \text{Entropy} = & (0.024.\log 0.024) + (0.071.\log 0.071) + (0.071.\log 0.071) + (0.048.\log \\ & 0.048) + (0.024.\log 0.024) + (0.048.\log 0.048) + (0.024.\log 0.024) + (0.143.\log \\ & 0.143) + (0.143.\log 0.143) + (0.214.\log 0.214) + (0.048.\log 0.048) + (0.095.\log \\ & 0.095) + (0.048.\log 0.048) \end{aligned}$$

$$\begin{aligned} \text{Entropy} = & 0.024*(-1.6197) + 0.071*(-1.1487) + 0.071*(-1.1487) + 0.048*(- \\ & 1.3187) + 0.024*(-1.6197) + 0.048*(-1.3187) + 0.024*(-1.6197) + 0.143*(-0.8446) \\ & + 0.143*(-0.8446) + 0.214*(-0.6695) + 0.048*(-1.3187) + 0.095*(-1.0222) + \\ & 0.048*(-1.3187) \end{aligned}$$

$$\begin{aligned} \text{Entropy} = & (-0.0388728) + (-0.815577) + (-0.815577) + (-0.0632976) + (0.0388728) \\ & + (-0.0632976) + (-0.0388728) + (-0.1207778) + (-0.1207778) + (-0.143273) + (- \\ & 0.0632976) + (-0.097109) + (-0.0632976) \end{aligned}$$

$$\text{Entropy} = \mathbf{-2.4829004}$$

4. 2. 3. Klasifikasi

Klasifikasi pada data yang di gunakan yaitu berupa 90 dataset data Training dan 10 data Testing, yaitu terdiri dari 45 gambar ikan busuk dan 45 gambar ikan segar dari ikan cakalang, dari 90 dataset data training ikan cakalang yang mana dari masing-masing gambar tersebut mempunyai nilai kontras, homogeneity, entropy, dan energy. Seperti pada tabel 4.13 dibawah ini:

Tabel 4.13 Hasil Perhitungan Data Training

No.	Data Training	Con(X1)	Hom(X2)	Ent(X3)	Ene(X4)	Y
0	 Gambar 0	194.5976177	0.48212288	0.69266220	4.5692884	IKAN
		6980068	08398003	68938836	24674708	BUSUK
1	 Gambar 1	189.4986230	0.47703152	0.69146775	4.6014087	IKAN
		973941	40687659	82641062	23731126	BUSUK
2	 Gambar 2	194.1855246	0.47837441	0.69520331	4.5825787	IKAN
		5492184	50989005	93679937	46335984	BUSUK
3	 Gambar 3	205.4232130	0.48372132	0.69223087	4.5717718	IKAN
		3247704	557768344	1407191	6880025	BUSUK
4	 Gambar 4	204.8721385	0.47492984	0.68639669	4.6359871	IKAN
		0927466	215012846	41142726	37304424	BUSUK

...	...	...	...	...	...	...
85	 Gambar 85	90.88807908 452029	0.71323367 78026995	0.82592071 8328623	2.7334158 77853284	IKAN SEGAR
86	 Gambar 86	79.43486305 407245	0.71537952 89964513	0.83057043 6871177	2.6882364 5606259	IKAN SEGAR
87	 Gambar 87	74.80077845 469783	0.71559451 31406959	0.83329928 11147496	2.6701974 02955136	IKAN SEGAR
88	 Gambar 88	94.56448793 660249	0.71227435 01843996	0.82363392 68833905	2.7530235 87456379 8	IKAN SEGAR
89	 Gambar 89	89.56769938 99751	0.71507924 93908011	0.82687995 00699494	2.7134578 68718750 7	IKAN SEGAR

0		143.4880307 0714817	143.488030 70714817	143.488030 70714817	143.48803 07071481 7	?
	Gambar Testing					

Pada tabel 4.14 ini yaitu hasil nilai dari 10 dataset tersebut di dapatkan dengan jarak 1 dan arah 0^0 .

Tabel 4.14 Hasil Perhitungan Data Testing

No.	Data Testing	X1	X2	X3	X4	Y
0	 IKAN BUSUK	89.689091 13979534	0.73245452 95983149	0.84743712 74045289	2.54162371 05378965	IKAN BUSUK
1	 IKAN BUSUK	89.142909 19842259	0.73097918 67309077	0.84581999 0676427	2.55457397 1026671	IKAN BUSUK
2	 IKAN BUSUK	88.373038 23149304	0.73391432 68036384	0.84875621 91509499	2.52691694 37778367	IKAN BUSUK

3	 IKAN BUSUK	93.832804 32994037	0.73284114 2987358	0.84636016 8771858	2.54565202 65485443	IKAN SEGAR
4	 IKAN BUSUK	92.398612 44578194	0.73222676 45496277	0.84530411 91163332	2.55211009 80185305	IKAN SEGAR
5	 IKAN SEGAR	143.48803 07071481 7	0.49514443 14260586	0.71575114 38246136	4.34717725 3436295	IKAN BUSUK
6	 IKAN SEGAR	133.23176 06140145 8	0.48886837 754305174	0.71686056 45914309	4.36251739 90002635	IKAN SEGAR
7	 IKAN SEGAR	95.682420 97250231	0.69774532 34846603	0.82142485 9924879	2.84547515 7077634	IKAN SEGAR

8		93.530303	0.68884075	0.81884913	2.90169436	IKAN
		33902143	90856093	53010083	89367114	SEGAR
	IKAN SEGAR					
9		95.818348	0.53665905	0.76015505	3.91779642	IKAN
		92356895	94834652	68866503	57798608	SEGAR
	IKAN SEGAR					

Berikut ini adalah tahapan untuk menghitung nilai Euclidean distance dan menentukan nilai k. tahapan-tahapan tersebut seperti di bawah ini :

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.

Euclidean Distance

$$\begin{aligned}
 d &= \sqrt{(194.5976 - 133.2317)^2 + (0.4821 - 0.4888)^2 + (0.6926 - 0.7168)^2 + (4.5692 - 4.3625)^2} \\
 &= \sqrt{3,765.7736 + 0.00004 + 0.00058 + 0.04272} \\
 &= \sqrt{3,8091136}
 \end{aligned}$$

= 1.95169

Tabel 4.15 Perbandingan Nilai

Data Training	Perhitungan jarak <i>Euclidean Distance</i>	Nilai
0	$d = \sqrt{3,765.7736 + 0.00004 + 0.00058 + 0.04272}$	61.3662525
1	$d = \sqrt{3.165,9640 + 0.00013 + 0.00064 + 0.05707}$	56.2674137
2	$d = \sqrt{3.715.36573 + 0.00011 + 0.00046 + 0.0484}$	60.9542017
3	$d = \sqrt{5,211.61267 + 0.000026 + 0.00060 + 0.04376}$	72.1918074
4	$d = \sqrt{5,132.34691 + 0.00019 + 0.00093 + 0.07474}$	71.6409294
...	...	...
85	$d = \sqrt{1,792.98893 + 0.05035 + 0.01190 + 2.65396}$	42.3757612
86	$d = \sqrt{2,894.10645 + 0.05130 + 0.01292 + 2.80328}$	53.8235446
87	$d = \sqrt{3,414.18176 + 0.05139 + 0.01354 + 2.86421}$	58.4560596
88	$d = \sqrt{1,495.16009 + 0.04990 + 0.01140 + 2.59049}$	38.7015746
89	$d = \sqrt{1,906.55363 + 0.05116 + 0.0121 + 2.71953}$	43.6959543

Tabel 4.16 Hasil Pengurutan dan Nilai Terkecil Berdasarkan Perhitungan Jarak dari Masing-masing Data Training ke Data Testing

Data	Euclidean	Y
88	38.70157	IKAN SEGAR
85	42.37576	IKAN SEGAR
89	43.69595	IKAN SEGAR
86	53.82354	IKAN SEGAR
1	56.26741	IKAN BUSUK
...	...	...
87	58.45605	IKAN SEGAR
2	60.95420	IKAN BUSUK
0	61.36625	IKAN BUSUK
4	71.64092	IKAN BUSUK
3	72.19180	IKAN BUSUK

Tabel 4.17 Hasil Jumlah K yang Sesuai Data

DATA	Euclidean	Y
88	38.70157	IKAN SEGAR
89	43.69595	IKAN SEGAR
86	53.82354	IKAN SEGAR
1	56.26741	IKAN BUSUK
87	58.45605	IKAN SEGAR
2	60.95420	IKAN BUSUK
0	61.36625	IKAN BUSUK

Jumlah K = 7 Sesuai Yang Ditentukan Sebelumnya

Y	Jumlah Data
IKAN BUSUK	3

IKAN SEGAR	4
------------	---

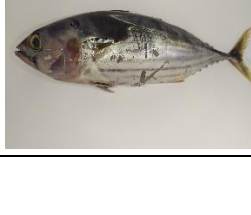
Tetangga Paling Banyak Muncul = Ikan Segar

4. 3. Evaluasi

4. 3. 1. Confusion Matrix

Pada penelitian ini menggunakan confusion Matrix sebagai metode. Dalam perhitungan akurasi pada penerapan deteksi kualitas ikan cakalang menggunakan tekstur warna. Evaluasi kinerja deteksi kualitas ikan cakalang di dasarkan dari jumlah pengujian objek.

Tabel 4.18 Hasil Data Testing

Image Testing	Data Aktual	Hasil Klasifikasi
	Ikan Busuk	Ikan Busuk
	Ikan Busuk	Ikan Busuk
	Ikan Busuk	Ikan Busuk
	Ikan Busuk	Ikan Segar
	Ikan Busuk	Ikan Segar

	Ikan Segar	Ikan Busuk
	Ikan Segar	Ikan Segar
	Ikan Segar	Ikan Segar
	Ikan Segar	Ikan Segar
	Ikan Segar	Ikan Segar

Tabel 4.19 Hasil Data Uji

	Ikan Busuk	Ikan Segar
Ikan Busuk	3	2
Ikan Segar	1	4

Jumlah data yang diklasifikasi dengan benar = 7

Jumlah data yang diklasifikasi salah = 3

Akurasi = (Data Uji Dikenali)/(Data uji) x 100 % Akurasi = 7/10 x 100
% = 70 %

BAB V

PEMBAHASAN

5. 1. Pembahasan Model

Pada percobaan model K-NN, dengan jumlah $K = 1$, dan Arah GLCM = 0° , dengan jarak = 1, maka mendapatkan nilai akurasi tertinggi Sebesar = 70%, dengan data training = 90, yaitu data ikan segar = 45, dan data ikan busuk = 45, Data Testing = 10.

Sedangkan pada percobaan model K-NN, dengan jumlah $K = 2$, dan Arah GLCM = 135° , dengan jarak = 2, maka mendapatkan nilai akurasi tertinggi sebesar = 60%, dengan Data Training = 90, yaitu data ikan segar = 45, dan data ikan busuk = 45, Data Testing = 10.

Pada percobaan model K-NN, dengan jumlah $K = 3$, dan Arah GLCM = 135° , dengan jarak = 1, Maka Mendapatkan Nilai Akurasi Sebesar = 60%, dengan Data Training = 90, yaitu data ikan segar = 45, dan data ikan busuk = 45, Data Testing = 10.

Pada percobaan model K-NN, dengan jumlah $K = 4$, dan Arah GLCM = 90° , dengan jarak = 4, Maka Mendapatkan Nilai Akurasi Sebesar = 80%, dengan Data Training = 90, yaitu data ikan segar = 45, dan data ikan busuk = 45, Data Testing = 10.

Pada percobaan dengan jumlah $K = 4$, dan Arah GLCM = 45° , dengan jarak = 4, maka mendapatkan nilai akurasi tertinggi Sebesar = 50%, dengan data training = 90, yaitu data ikan segar = 45, dan data ikan busuk = 45, Data Testing = 10.

Sedangkan pada percobaan dengan jumlah $K = 3$, dan Arah GLCM = 45° , dengan jarak = 3, maka mendapatkan nilai akurasi tertinggi sebesar = 70%, dengan Data Training = 90, yaitu data ikan segar = 45, dan data ikan busuk = 45, Data Testing = 10.

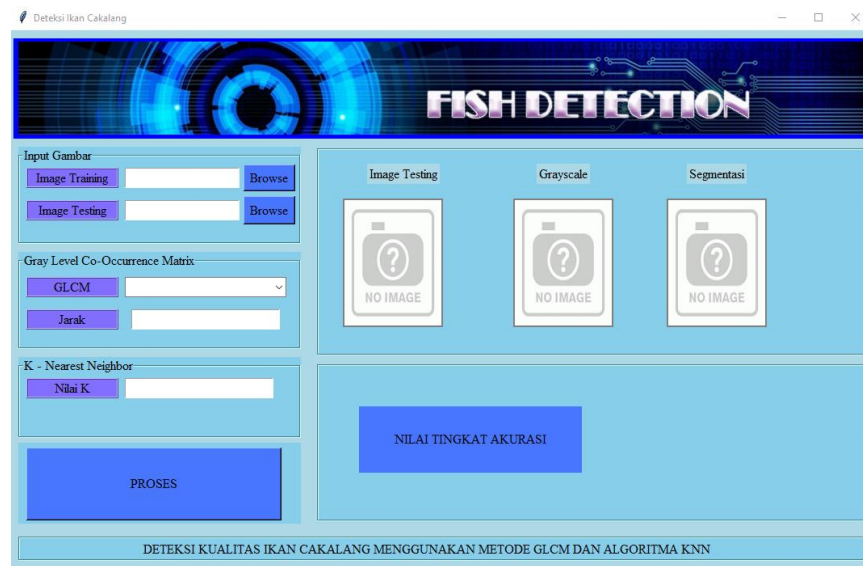
Dan dengan jumlah $K = 1$, dan Arah GLCM = 45° , dengan jarak = 1, Maka Mendapatkan Nilai Akurasi Sebesar = 50%, dengan Data Training = 90, yaitu data ikan segar = 45, dan data ikan busuk = 45, Data Testing = 10.

Pada percobaan model K-NN, dengan jumlah $K = 2$, dan Arah GLCM = 90° , dengan jarak = 2, Maka Mendapatkan Nilai Akurasi Sebesar = 40%, dengan Data Training = 90, yaitu data ikan segar = 45, dan data ikan busuk = 45, Data Testing = 10.

Pada tahap pengujian metode terdapat kendala pada tahap pengenalan kualitas, karena penggunaan sampel ikan harus diperhatikan dari kontras dan posisi dari data sampel, karenan ada beberapa sampel data yang tidak biasa dikenali atau tidak terdeteksi. Arah dan jarak GLCM sangat berpengaruh untuk pengenalan kualitas ikan, hasil yang di dapatkan untuk metode GLCM dengan K-NN dengan arah 80° dan jarak 4 lebih tinggi tingkat akurasi. Hasil yang di dapatkan dari pengujian metode akurasi yang diperoleh sudah cukup baik, hasil dari perhitungan akurasi di atas bahwa semakin banyak nilai K semakin tinggi nilai akurasi yang diperoleh, dan sebaliknya. jumlah K 4 lebih tinggi tingkat akurasi kualitas ikan cakalang menggunakan tekstur warna, ketimbang jumlah K 2. Dari hasil perhitungan akurasi untuk pengujian Klasifikasi mendapatkan nilai sebesar 80%. Pada penelitian ini penggunaan metode K-NN kurang baik untuk tahap klasifikasi.

5. 2. Pembahasan Sistem

1. Tampilan Menu Utama



Gambar 5.1 Tampilan Menu Utama

Tampilan pertama saat program dijalankan

2. Tampilan Browse Data Training

Halaman ini adalah tampilan jika pengguna menekan tombol browse untuk data training.

3. Tampilan Browse Data Testing

Halaman ini adalah tampilan jika pengguna menekan tombol browse untuk data testing.

Tampilan ini akan muncul saat pengguna memilih/mengklik *browser* pada *Image Testing*, dan pengguna akan memilih folder yang tersimpan gambar.

4. Tampilan Setelah Pemilihan Folder *Traning* dan data *Testing*

Tampilan ini akan memunculkan nama folder yang telah dipilih pengguna.

5. Tampilan Pemasukan Nilai GLCM

Tampilan ini akan muncul saat pengguna telah memasukan nilai dan memilih perhitungan pada GLCM.

6. Tampilan Pemasukan Nilai K

Tampilan ini akan muncul saat pengguna telah memasukan nilai K.

7. Tampilan Hasil Kualitas Ikan Cakalang

Tampilan ini akan muncul saat pengguna mengklik/memilih tombol atau button proses setelah menginput data training, data testing, arah dan jarak GLCM serta nilai k. Tampilan ini adalah tampilan hasil pemrosesan.

8. Tampilan Hasil Klasifikasi Dan Nilai Akurasi

Tampilan ini akan muncul bersamaan dengan tampilan hasil kualitas ikan cakalang.

BAB VI

PENUTUP

6. 1. Kesimpulan

Berdasarkan rumusan masalah, maka tujuan penelitian ini yaitu Mengetahui hasil penerapan GLCM dan K-NN untuk proses deteksi kualitas ikan cakalang. Menggunakan tekstur warna dalam tingkata akurasi kesegarannya maka dapat ditarik suatu kesimpulan bahwa Pada percobaan model K-NN, dengan jumlah $K = 1$, dan Arah GLCM = 0° , dengan jarak = 1, maka mendapatkan nilai akurasi tertinggi Sebesar = 70%. Sedangkan pada percobaan model K-NN, dengan jumlah $K = 2$, dan Arah GLCM = 135° , dengan jarak = 2, maka mendapatkan nilai akurasi tertinggi sebesar = 60%. Pada percobaan model K-NN, dengan jumlah $K = 3$, dan Arah GLCM = 135° , dengan jarak = 1, Maka Mendapatkan Nilai Akurasi Sebesar = 60%. Pada percobaan model K-NN, dengan jumlah $K = 4$, dan Arah GLCM = 90° , dengan jarak = 4, Maka Mendapatkan Nilai Akurasi Sebesar = 80%. Pada percobaan dengan jumlah $K = 4$, dan Arah GLCM = 45° , dengan jarak = 4, maka mendapatkan nilai akurasi tertinggi Sebesar = 50%. Sedangkan pada percobaan dengan jumlah $K = 3$, dan Arah GLCM = 45° , dengan jarak = 3, maka mendapatkan nilai akurasi tertinggi sebesar = 70%. Dan dengan jumlah $K = 1$, dan Arah GLCM = 45° , dengan jarak = 1, Maka Mendapatkan Nilai Akurasi Sebesar = 50%. Pada percobaan model K-NN, dengan jumlah $K = 2$, dan Arah GLCM = 90° , dengan jarak = 2, Maka Mendapatkan Nilai Akurasi Sebesar = 40%.

6. 2. Saran

Setelah melakukan Penelitian deteksi kualitas ikan cakalang, ada beberapa saran yang perlu di perhatikan untuk mencapai tujuan yang di harapkan, yaitu sebagai berikut:

1. Penulis berharap untuk penelitian selanjutnya agar mendapatkan hasil akurasi yang lebih baik.
2. Penulis berharap agar hasil segmentasi berikutnya menjadi lebih baik lagi.
3. Penulis berharap untuk penelitian selanjutnya agar bisa membedakan ikan yang sudah lecet-lecet masih segar atau sudah busuk

DAFTAR PUSTAKA

- [1]. Altien J.Rindengan, Mans Mananohas, “Perancangan Sistem Penentuan Tingkat Kesegaran Ikan Cakalang Menggunakan Metode *Curve Fitting* Berbasis Citra Digital Mata Ikan” Jurnal Ilmiah Sains Vol.17 No.2 Oktober 2017.
- [2]. Christina Litaay, Sugeng Hari Wisudo, John Haluan, dan Bambang Harianto “Pengaruh Perbedaan Metode Pendinginan Dan Waktu Penyimpanan Terhadap Mutu Organoleptik Ikan Cakalang Segar” Jurnal Ilmu dan Teknologi Kelautan Tropis, Vol.9, No.2 Desember 2017.
- [3]. Christina Litaay, Sugeng Hari Wisudo, John Haluan, dan Bambang Harianto “Pengaruh Perbedaan Metode Pendinginan Dan Waktu Penyimpanan Terhadap Mutu Organoleptik Ikan Cakalang Segar” Jurnal Ilmu dan Teknologi Kelautan Tropis, Vol.9, No.2 Desember 2017.
- [4]. Christina Litaay, Sugeng Hari Wisudo, John Haluan, dan Bambang Harianto “Pengaruh Perbedaan Metode Pendinginan Dan Waktu Penyimpanan Terhadap Mutu Organoleptik Ikan Cakalang Segar” Jurnal Ilmu dan Teknologi Kelautan Tropis, Vol.9, No.2 Desember 2017.
- [5]. Christina Litaay, Sugeng Hari Wisudo, John Haluan, dan Bambang Harianto “Pengaruh Perbedaan Metode Pendinginan Dan Waktu Penyimpanan Terhadap Mutu Organoleptik Ikan Cakalang Segar” Jurnal Ilmu dan Teknologi Kelautan Tropis, Vol.9, No.2 Desember 2017.
- [6]. Altien J.Rindengan, Mans Mananohas, “Perancangan Sistem Penentuan Tingkat Kesegaran Ikan Cakalang Menggunakan Metode *Curve Fitting* Berbasis Citra Digital Mata Ikan” Jurnal Ilmiah Sains Vol.17 No.2 Oktober 2017.
- [7]. Muhammad Jumnahdi, Masayu Anisah, Sabilal Rasyad, “Aplikasi Matrik Kookuransi Tingkat Keabuan Untuk Analisa Citra IkanTenggiri” Jurnal Ampere, Volume.3 No.1 Juni 2018.

[8].Refta Listia, Agus Harjoko, “Klasifikasi Massa pada Citra Mammogram Berdasarkan

Gray Level Cooccurrence Matrix (GLCM)” IJCS, Vol.8, No.1, Januari 2014.

[9]. Sumarlin, “Implementasi Algoritma K-Nearest Neighbor Sebagai Pendukung Keputusan Klasifikasi Penerima Beasiswa PPA dan BBM” Jurnal Sistem Informasi Bisnis, Vol.1, 11 Maret 2015.

[10]. 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” Jurnal Teknologi dan Sistem Komputer, Vol.6, No.2 14 Maret 2018.

[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 Cooccurrence 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, "Pengenal Motif Batik Menggunakan Metode Transformer Paket *Wavelet*," *Tugas Akhir Teknik Informatika*, no.Universitas Widyatama, Bandung, 2013.

[26]. Insanudin Andi, "Ekstraksi Informasi Kemacetan pada Media Digital," *Tugas Akhir Teknik Informatika*, no. Universitas Komputer Indonesia, Bandung, 2013.

KODE PROGRAM

1. Pembuatan Form

```
import tkinter as tk
import tkinter.filedialog as fdialog
import tkinter.ttk as ttk
from PIL import Image, ImageTk
#from sklearn.preprocessing import OrdinalEncoder
import dataTraining
import cv2

class Aplication(tk.Frame):
    def __init__(self, parent):
        self.parent = parent
        self.frames()
        self.widget()

    def frames(self):
        #frame judul
        self.FrameAtas = tk.Frame(self.parent, width=1058, height=150,
background="light blue", relief='raised')
        self.FrameAtas.grid(row=0, column=0, columnspan=3, padx=4,
pady=10, sticky="NSEW")
        self.FrameAtas.grid_propagate(0)

        #frame input gambar
        self.FrameKiri1 = tk.LabelFrame(self.parent, height=120,
background="sky blue", text="Input Gambar", font="Times", fg="black")
        self.FrameKiri1.grid(row=1, column=0, padx=10, sticky="NSEW")
        self.FrameKiri1.grid_propagate(0)

        #glcm
        self.FrameKiri2 = tk.LabelFrame(self.parent, height=120,
background="sky blue", text="Gray Level Co-Occurrence Matrix",
font="Times", fg="black")
        self.FrameKiri2.grid(row=2, column=0, padx=10, pady=10,
sticky="NSEW")
        self.FrameKiri2.grid_propagate(0)

        #knn
```

```

        self.FrameKiri3 = tk.LabelFrame(self.parent, height=100,
background="sky blue", text="K - Nearest Neighbor", font="Times",
fg="black")
        self.FrameKiri3.grid(row=3, column=0, padx=10, sticky="NSEW")
        self.FrameKiri3.grid_propagate(0)

        #proses
        self.FrameKiri4 = tk.Frame(self.parent, height=100, background="sky
blue")
        self.FrameKiri4.grid(row=5, column=0, padx=10, pady=6,
sticky="NSEW")
        self.FrameKiri4.grid_propagate(0)

        #hasil gambar
        self.FrameKanan5 = tk.LabelFrame(self.parent, background="sky blue",
font="Times", fg="black")
        self.FrameKanan5.grid(row=1, column=1, rowspan=2, columnspan=2,
padx=10, pady=2, sticky="NSEW")
        self.FrameKanan5.grid_propagate(0)

        #hasil akhir
        self.FrameKanan6 = tk.LabelFrame(self.parent, background="sky blue",
font="Times", fg="gray")
        self.FrameKanan6.grid(row=3, column=1, rowspan=3, columnspan=2,
padx=10, pady=9, sticky="NSEW")
        self.FrameKanan6.grid_propagate(0)

        #kaki
        self.FrameBawah = tk.LabelFrame(self.parent, height=30, width=1020,
background="sky blue")
        self.FrameBawah.grid(row=6, column=0, columnspan=3, padx=10,
pady=10, sticky="NSEW")
        self.FrameBawah.grid_propagate(0)

    def widget(self):

        #frame
        #logo
        self.PathTestingImg = 'D:\SKRIPSI HAMDAN\Banner\Banner6.jpg'
        self.imT = Image.open(self.PathTestingImg)
        self.imT = ImageTk.PhotoImage(self.imT)
        self.label1 = tk.Label(self.FrameAtas, width=1058, height=120,
bg="blue", image=self.imT)

```

```

self.label1.pack(side="top")

#frame 1
#label data training
self.label1 = tk.Label(self.FrameKiri1, text="Image Training",
font="Times", relief="ridge", width=12, background="slateblue1")
self.label1.grid(row=3, column=0, sticky="NSEW", padx=8, pady=5)

#textbox data training
self.nilai = tk.StringVar(self.FrameKiri1, value="")
self.txtrain = tk.Entry(self.FrameKiri1, textvariable = self.nilai,
width=23)
self.txtrain.grid(row=3, column=1, sticky="NSEW", pady=5)

#button browse training
self.path2 = tk.Button(self.FrameKiri1, text="Browse", font="Times",
width=6, background="Royalblue1",
command=lambda:self.nilai.set(fdialog.askdirectory()))
self.path2.grid(row=3, column=5, padx=5)

#label data testing
self.label2 = tk.Label(self.FrameKiri1, text="Image Testing",
font="Times", relief="ridge", background="slateblue1")
self.label2.grid(row=4, column=0, sticky="NSEW", padx=8, pady=10)

#textbox data testing
self.nilai1 = tk.StringVar(self.FrameKiri1, value="")
self.txtest = tk.Entry(self.FrameKiri1, textvariable = self.nilai1)
self.txtest.grid(row=4, column=1, sticky="NSEW", pady=10)

#button browse testing
self.path1 = tk.Button(self.FrameKiri1, text="Browse",
font="Times",background="Royalblue1",
command=lambda:self.nilai1.set(fdialog.askopenfilename()))
self.path1.grid(row=4, column=5, sticky="NSEW", padx=5, pady=5)

#frame 2
#COMBOBOX GLCM
arah = tk.IntVar(self.FrameKiri2)

```

```

        self.label = tk.Label(self.FrameKiri2, text="GLCM", width=12,
font="Times", relief="ridge", background="slateblue1")
        self.label.grid(row=5, column=0, padx=8 , pady=10)

        #arah GLCM
        arah = tk.StringVar(self.FrameKiri2, value="")
        self.cbarahglcm = ttk.Combobox(self.FrameKiri2, textvariable = arah,
width=10)
        self.cbarahglcm['value'] = (0, 45, 90, 135)
        self.cbarahglcm.current(0)
        self.cbarahglcm.grid(row=5, column=1, sticky="NSEW", pady=10)

        #jarak GLCM
        self.label1 = tk.Label(self.FrameKiri2, text="Jarak", font="Times",
relief="ridge", width=12, background="slateblue1")
        self.label1.grid(row=6, column=0, sticky="NSEW", padx=8, pady=5)

        jarak = tk.StringVar(self.FrameKiri2, value="")
        self.tbjarak = tk.Entry(self. FrameKiri2, textvariable = jarak, width=30)
        self.tbjarak.grid(row=6, column=1, sticky="NSEW", padx=8, pady=5 )


        #frame 3
        #klasifikasi KNN
        #label iterasi
        self.label = tk.Label(self.FrameKiri3, text="Nilai K", font="Times",
relief="ridge", width=12, background="slateblue1")
        self.label.grid(row=0, column=0, sticky="NSEW", padx=8, pady=5)

        #textbox iterasi
        nilai2 = tk.StringVar(self.FrameKiri3, value="")
        self.textboxKnn = tk.Entry(self.FrameKiri3, textvariable = nilai2,
width=30)
        self.textboxKnn.grid(row=0, column=1, sticky="NSEW", pady=5)


        #frame 4
        #button proses
        self.path2 = tk.Button(self.FrameKiri4, width=34, height=4,
text="PROSES",
font="Times",background="Royalblue1",command=self.Proses)

```

```

self.path2.grid(row=5, column=0, padx=10, pady=6, sticky="NSEW")

#frame 5
#output
#label hasil Image Asli
self.label = tk.Label(self.FrameKanan5, text="Image Testing",
font="Times", background="light blue", fg="black")
self.label.grid(row=0, column=0, padx=60, pady=17)
#images Hasil Testing
self.noimage1 = 'D:/SKRIPSI HAMDAN/Banner/noimage.jpg'
self.noimageTest1 = Image.open(self.noimage1)
self.noimageTest1 = ImageTk.PhotoImage(self.noimageTest1)
self.labelTest1 = tk.Label(self.FrameKanan5, bg = "gray",
image=self.noimageTest1)
self.labelTest1.grid(row=1, column=0, padx=30, pady=1, sticky="W")

#label Grayscale
self.label = tk.Label(self.FrameKanan5, text="Grayscale", font="Times",
background="light blue", fg="black")
self.label.grid(row=0, column=1, padx=60, pady=17)
#images Hasil Grayscale
self.noimage2 = 'D:/SKRIPSI HAMDAN/Banner/noimage.jpg'
self.noimageTest2 = Image.open(self.noimage2)
self.noimageTest2 = ImageTk.PhotoImage(self.noimageTest2)
self.labelTest2 = tk.Label(self.FrameKanan5, bg = "gray",
image=self.noimageTest2)
self.labelTest2.grid(row=1, column=1, padx=30, pady=1, sticky="N")

#label Segmentasi
self.label = tk.Label(self.FrameKanan5, text="Segmentasi",
font="Times", background="light blue", fg="black")
self.label.grid(row=0, column=2, padx=60, pady=17)
#images Hasil Segmentasi
self.noimage3 = 'D:/SKRIPSI HAMDAN/Banner/noimage.jpg'
self.noimageTest3 = Image.open(self.noimage3)
self.noimageTest3 = ImageTk.PhotoImage(self.noimageTest3)
self.labelTest3 = tk.Label(self.FrameKanan5, bg = "gray",
image=self.noimageTest3)
self.labelTest3.grid(row=1, column=2, padx=30, pady=1, sticky="S")

```

```

#frame 6
#label keterangan hasil output
self.label12 = tk.Label(self.FrameKanan6, text="NILAI TINGKAT
AKURASI", width=30, height=4, font="Times", background="Royalblue1")
self.label12.grid(row=2, column=0, columnspan=5, padx=50, pady=50)

```

```

#frame 7
#label kaki
self.label = tk.Label(self.FrameBawah, text="DETEKSI KUALITAS
IKAN CAKALANG MENGGUNAKAN METODE GLCM DAN
ALGORITMA KNN", font="Times", fg="black")
self.label.grid(row=2, column=2, sticky="NSEW", padx=150, pady=1)
self.label.config(background="sky blue")
self.FrameBawah.grid_propagate(0)

```

```

def Proses(self):
    #get image training
    folder_train = self.txtrain.get()
    #get image testing
    folder_test = self.txtest.get()
    #glcm
    GLCM= self.cbarahglcm.get()
    jarak = self.tbjarak.get()

    #textbox iterasi
    K = self.textboxKnn.get()

    #hasil output akhir
    a = dataTraining. analisiscitra()
    hasil, img_segmen, gray, = a.proses_rekognisi(folder_train, folder_test,
    GLCM, jarak,K)
    self.label12.configure(text = hasil)

    #hasil gambar
    #hasil image 1
    imgTest = Image.open(self.txtest.get())
    imgTest = imgTest.resize((165, 165), Image.ANTIALIAS)
    imgTest = ImageTk.PhotoImage(imgTest)
    self.labelTest1.configure(image=imgTest)
    self.labelTest1.image_names = imgTest

```

```
#hasil image 2
imgGray = Image.fromarray(gray)
imgGray = imgGray.resize((165, 165), Image.ANTIALIAS)
imgGray = ImageTk.PhotoImage(imgGray)
self.labelTest2.configure(image=imgGray)
self.labelTest2.image_names = imgGray
```

```
#hasil image 3
imgsegmen = cv2.cvtColor(img_segmen, cv2.COLOR_BGR2RGB)
imgsegmen = Image.fromarray(imgsegmen)
imgsegmen = imgsegmen.resize((165,165), Image.ANTIALIAS)
imgsegmen = ImageTk.PhotoImage(imgsegmen)
self.labelTest3.configure(image=imgsegmen)
self.labelTest3.image_names = imgsegmen
```

```
def main():
    root = tk.Tk()
    root.geometry('1070x670+130+10')
    root.configure(background='light blue')
    app = Aplication(root)
    app.parent.title("Deteksi Ikan Cakalang")
    root.mainloop()
```

```
if __name__ == '__main__':
    main()
```


2. Image Processing

```
import cv2
import numpy as np
from skimage import feature
from scipy.stats import entropy
from imutils import paths
import os
from scipy import stats
from sklearn.neighbors import KNeighborsClassifier

class analisiscitra:
    def segmentasi(self, img):
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        blur = cv2.GaussianBlur(gray,(5,5),0)
        ret, alpha = cv2.threshold(np.uint8(blur), 0, 255,
cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
        alpha[alpha==255] = 1
        rect = cv2.boundingRect(alpha)
        r = img[:, :, 0] * alpha
        g = img[:, :, 1] * alpha
        b = img[:, :, 2] * alpha
        hasil = cv2.merge((r,g,b))
        x,y,w,h = rect
        result = hasil[y:y+h, x:x+w]
        return result, gray

    def fitur_glcmlcm(self, img, GLCM, jarak):
        img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        matrix_glcmlcm = feature.greycomatrix(img, [jarak], [GLCM],
symmetric=False, normed=True)
        contrast = feature.greycomprops(matrix_glcmlcm, 'contrast')[0,0]
        energy = feature.greycomprops(matrix_glcmlcm, 'energy')[0,0]
        homogeneity = feature.greycomprops(matrix_glcmlcm, 'homogeneity')[0,0]
        ent = entropy(matrix_glcmlcm.flatten())
        hasil_fitur = [contrast, energy, homogeneity, ent]
        return hasil_fitur

    def get_Img_Path(self, folderPath):
        imPath = []
        for path in paths.list_images(folderPath):
            imPath.append(path)
        return imPath
```

```

def get_label(self, folderPath):
    label = []
    imPath = self.get_Img_Path(folderPath)
    for item in imPath:
        label.append(os.path.split(os.path.dirname(item))[-1])
    return label

def training_data(self, folderPath, GLCM, jarak):
    pathList = self.get_Img_Path(folderPath)
    images = []
    for imPath in pathList:
        img = cv2.imread(imPath)
        images.append(img)
    data_train = []
    for img in images:
        hasil_segmen, gray = self.segmentasi(img)
        ekstraksi = self.fitur_glcml(hasil_segmen, GLCM, jarak)
        data_train.append(ekstraksi)
    label_train = self.get_label(folderPath)
    return data_train, label_train

def testing_data(self, img, GLCM, jarak):
    data_test = []
    img = cv2.imread(img)
    hasil_segmen, gray = self.segmentasi(img)
    hasil_ekstrak_glcml = self.fitur_glcml(hasil_segmen, GLCM, jarak)
    data_test.append(hasil_ekstrak_glcml)
    return data_test, hasil_segmen, gray

def proses_rekognisi(self, folder_train, folder_test, GLCM, jarak, K):
    x_train, y_train = self.training_data(folder_train, GLCM, jarak)
    x_test, hasil_segmen, gray = self.testing_data(folder_test, GLCM, jarak)
    # allData = self.encode_data(data_train + data_test)
    # x_train = allData[0:-int(len(data_test))]
    # x_test = allData[-int(len(data_test)):]
    clf = KNeighborsClassifier(n_neighbors=int(K), weights='uniform',
                              algorithm='auto', metric='euclidean')
    clf.fit(x_train, y_train)
    final_hasil = clf.predict(x_test)
    return final_hasil, hasil_segmen, gray

#def main():

```

```
# a = analisiscitra()
# folder_train = 'D:\SKRIPSI HAMDAN\DataTraining Ikan Cakalang'
# img_test = 'Ikan Busuk.jpg'
# result = a.proses_rekognisi(folder_train, img_test, GLCM, jarak,K)
# print(result)
# cv2.waitKey(0)
# cv2.destroyAllWindows()
# if __name__ == '__main__':
#     main()
```

DAFTAR RIWAYAT HIDUP



Nama : Hamdan Pakaya

Tempat,Tgl Lahir : Gorontalo, 21 Oktober 1997

Pekerjaan : Mahasiswa

Email : hamdanpakaya456@gmail.com

Daftar Riwayat Hidup :

1. Tahun 2011, menyelesaikan Pendidikan di Sekolah Dasar Negeri 101 Kota Utara Sekarang SDN 10 Kota Utara, Kecamatan Kota Utara, Kota Gorontalo.
2. Tahun 2013, menyelesaikan Pendidikan di Sekolah Menengah Pertama Negeri 12 Kota Utara, Kecamatan Kota Utara, Kota Gorontalo.
3. Tahun 2016, menyelesaikan Pendidikan di Sekolah Menenga Kejuruan Negeri 3 Gorontalo, Kecamatan Kota Tengah, Kota Gorontalo.
4. Tahun 2016, telah diterima menjadi Mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo