

**MENGIDENTIFIKASI KUALITAS UDANG SEGAR
MENGUNAKAN METODE *GRAY LEVEL*
CO-occurrence MATRIX DAN
ARTIFICIAL NEURAL
*NETWORK***

Oleh
WANDA APRILIA PANGEMANAN
T3115115

SKRIPSI

Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar sarjana



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

PERSETUJUAN

**MENGIDENTIFIKASI KUALITAS UDANG SEGAR
MENGUNAKAN METODE *GRAY LEVEL*
CO-occurrence MATRIX DAN
ARTIFICIAL NEURAL
*NETWORK***

Oleh

WANDA APRILIA PANGEMANAN

T3115115

SKRIPSI

Untuk memenuhi salah satu syarat Ujian Akhir guna memperoleh gelar
Sarjana Komputer, ini telah disetujui oleh Tim Pembimbing

Gorontalo, Juni 2022

Pembimbing I



Irma Surya Kumala Idris, M.Kom
NIDN.0921128801

Pembimbing II



Sudirman S. Panna, M.Kom
NIDN.0924038205

LEMBAR PENGESAHAN SKRIPSI
MENGIDENTIFIKASI KUALITAS UDANG SEGAR
MENGGUNAKAN METODE GRAY LEVEL
CO-occurrence MATRIX DAN
ARTIFICIAL NEURAL
NETWORK

OLEH

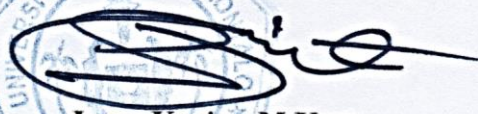
WANDA APRILIA PANGEMANAN
T3115115

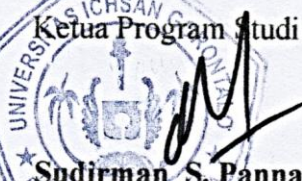
Di Periksa oleh Panitia Ujian Strata Satu(S1)
Universitas Ichsan Gorontalo
Gorontalo, Juni 2022

1. Ketua Penguji
Asmaul Husna N, M.Kom
2. Anggota Penguji
Abd Rahmat Karim Haba, M.Kom
3. Anggota Penguji
Andi Kamaruddin, M.Kom
4. Anggota Penguji
Irma Surya Kumala Idris, M.Kom
5. Anggota Penguji
Sudirman S. Panna, M.Kom

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

Mengetahui:

Dekan Fakultas Ilmu Komputer

Jorry Karim, M.Kom
NIDN.0918077302

Ketua Program Studi

Sudirman S. Panna, M.Kom
NIDN.0924038205

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, Juni 2022

Yang membuat pernyataan



Wanda Aprilia Pangemanan

ABSTRACT

WANDA APRILIA PANGEMANAN. T3115115. IDENTIFICATION OF FRESH SHRIMP QUALITY USING GRAY LEVEL CO-OCCURRENCE MATRIX AND ARTIFICIAL NEURAL NETWORK METHODS

This study is conducted to know the fresh shrimp quality. In this study, the data collection is through images of shrimp of a variety of sizes and with the number of classes. There are two classes, namely fresh and not fresh. This study is observed independently. The methods used in this study are the Gray Level Co-occurrence Matrix and Artificial Neural Network methods. The performance of using the GLCM and ANN methods in the identification process of fresh shrimp quality indicates a very good performance as proven by the accuracy of 93%, recall of 100%, the precision of 90%, and an F1 score of 95%.

Keywords: fresh shrimp quality, GLCM, ANN



ABSTRAK

WANDA APRILIA PANGEMANAN. T3115115. MENGIDENTIFIKASI KUALITAS UDANG SEGAR MENGGNAKAN METODE *GRAY LEVEL CO-occurrence MATRIX* DAN *ARTIFICIAL NEURAL NETWORK*

Penelitian ini dilakukan dengan tujuan untuk mengetahui kualitas udang segar. Dalam penelitian ini data yang dikumpulkan berupa gambar udang yang ukurannya bervariasi dengan jumlah kelas ada dua yang segar dan tidak segar. Penelitian ini diamati secara mandiri. Metode yang digunakan dalam penelitian ini adalah metode Gray Level Co-occurrence Matrix dan Artificial Neural Network. Kinerja penggunaan metode GLCM dan ANN dalam melakukan proses identifikasi kualitas udang memiliki kinerja yang sangat baik ditunjukkan dengan akurasi sebesar 93%, recall sebesar 100%, precission sebesar 90% dan nilai F1 score sebesar 95%.

Kata Kunci : Udang Segar, GLCM, ANN



KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan skripsi ini dengan judul “Mengidentifikasi Kualitas Udang Dilihat Dari Warnanya” 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. Untuk itu, dengan segala keikhlasan dan kerendahan hati, penulis, mengucapkan banyak terima kasih dan penghargaan yang setinggi-tingginya kepada :

1. Ibu Dr. Hj. Juriko Abdussamad, Msi, selaku Ketua Yayasan Pengembang Ilmu Pengetahuan dan Teknologi (YPIPT).
2. Bapak Dr. Abd. Gaffar Latjokke, M.Si, selaku Rektor Universitas Ichsan Gorontalo.
3. Bapak Jorry Karim, S.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.
4. Bapak Sudirman Melangi, S.Kom, M.Kom, selaku Wakil Dekan 1 Bidang Akademik Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.
5. Ibu Irma Surya Kumala Idris, M.Kom, selaku Wakil Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo, sekaligus sebagai Pembimbing I.
6. Bapak Sudirman S. Panna, S.Kom, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo, Sekaligus Sebagai Pembimbing II.
7. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis.
8. Kedua Orang Tua saya yang tercinta, atas segala kasih dan sayang, jerih payah dan Doa restunya dalam menyertai penulis.
9. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan, waktu dan dukungan yang sangat besar kepada penulis.

10. Kepada semua pihak yang ikut membantu dalam penyelesaian skripsi ini yang tak sempat penulis sebutkan satu-persatu.

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

Gorontalo, Juni 2022

Wanda Aprilia Pangemanan

DAFTAR ISI

PERSETUJUAN	ii
LEMBAR PENGESAHAN SKRIPSI.....	iii
PERNYATAAN SKRIPSI.....	iv
<i>ABSTRACT</i>	v
ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah	2
1.3 Rumusan Masalah	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB II LANDASAN TEORI	4
2.1 Tinjauan Studi	4
2.2 Tinjauan Pustaka	5
2.2.1 Udang Segar.....	5
2.2.2 Citra Digital	6
2.2.3 Computer Vision.....	7
2.2.4 Pengenalan Pola.....	7
2.2.5 Klasifikasi	8
2.2.6 Gray Level Co-occurrence Matriks	8
2.2.7 Artificial Neural Network.....	9
2.2.8 Confusion Matrix.....	13
2.4 Kerangka Pikir.....	15

BAB III METODE PENELITIAN.....	16
3.1 Jenis, Metode, Subjek, Objek, Waktu dan Lokasi Penelitian.....	16
3.2 Pengumpulan Data.....	16
3.3 Pemodelan	17
3.3.1 Pre-Processing	18
3.3.2 Ekstraksi Fitur Warna	18
3.3.3 Fitur Training.....	18
3.3.4 Fitur Testing.....	18
3.3.5 Model	18
3.4 Evaluasi Model	19
BAB IV HASIL PENELITIAN	20
4.1 Hasil pengumpulan data	20
4.2 Peningkatan kualitas citra.....	22
4.3 Segmentasi.....	24
4.4 Ekstraksi fitur menggunakan GLCM	24
4.5 Proses klasifikasi menggunakan ANN	31
4.5.1 Proses transformasi data	31
4.5.2 Training data	32
4.6 Hasil klasifikasi	36
4.7 Evaluasi model	38
4.7.1 Recall	38
4.7.2 Precision.....	39
4.7.3 Accuracy	39
4.7.4 F1 Score	40
BAB V PEMBAHASAN HASIL	41
5.1 Objek	41
5.2 Pembahasan hasil eksperimen	41
BAB VI PENUTUP	44
6.1 Kesimpulan.....	44
6.2 Saran.....	44
DAFTAR PUSTAKA	46

DAFTAR GAMBAR

Gambar 2.1	Konsep Dasar Pemodelan ANN.....	9
Gambar 2.2	Multilayer Perceptron.....	11
Gambar 2.3	Fungsi-fungsi Aktivasi	11
Gambar 3.1	Metode Penelitian.....	16
Gambar 4.1	Hasil pengumpulan data.....	20
Gambar 4.2	Filter 3 x 3	22
Gambar 4.3	Pixel Citra.....	22
Gambar 4.4	kernel/filter yang digunakan	22
Gambar 4.5	Matriks hasil proses morfologi.....	23
Gambar 4.6	Hasil akhir proses morfologi.....	23
Gambar 4.7	Nilai pixel citra hasil proses segmentasi	25
Gambar 4.8	Matrix glcm yang belum terisi	26
Gambar 4.9	Hasil proses pencarian matrix glcm	26
Gambar 4.10	Arsitektur ANN.....	32

DAFTAR TABEL

Tabel 2.1	Tinjauan Studi	4
Tabel 4.1	Hasil proses ekstraksi fitur pada data training	27
Tabel 4.2	Hasil transformasi data training	31
Tabel 4.3	Data training ke-1.....	33
Tabel 4.4	Nilai bobot Hidden Layer 1.....	34
Tabel 4.5	Nilai bobot Hidden Layer 2.....	35
Tabel 4.6	Nilai bobot Hidden layer 3.....	35
Tabel 4.7	Nilai bobot Output Layer	36
Tabel 4.8	Hasil proses klasifikasi menggunakan algoritma ANN	37
Tabel 4.9	Hasil Confussion Matrix	38

BAB I

PENDAHULUAN

1.1 Latar Belakang

Udang merupakan salah satu komoditas utama hasil perikanan indonesia, udang sebagai salah satu makanan yang dimanfaatkan oleh masyarakat karena udang mempunyai rasa yang enak dan kandungan protein yang tinggi. Segar atau tidaknya udang bisa mempengaruhi kandungan gizi dan rasanya. Udang segar bisa dilihat dari tekstur kulit dan dagingnya yang masih keras.

Terkadang dalam memilih kualitas udang segar yang baik dikonsumsi, kita belum begitu paham dalam membedakan kualitas udang segar dan udang tidak segar. Mengonsumsi udang yang sudah tidak segar dapat menimbulkan beberapa penyakit seperti diare dan keracunan. Untuk mengidentifikasi daging udang segar dan daging udang yang tidak segar memerlukan tenaga ahli dan waktu yang lama. Oleh karena itu diperlukan sebuah aplikasi yang dapat mengidentifikasi daging udang segar dan daging udang yang sudah tidak segar dengan cepat.

Salah satu yang dapat digunakan untuk mengatasi permasalahan di atas adalah dengan menggunakan teknologi Computer Vision agar menghindari adanya kecurangan dari pihak yang tak bertanggung jawab. Dalam penelitian ini dilakukan proses pengenalan objek diantaranya menggunakan metode GLCM dan ANN.

Salah satu metode Pengolahan Citra yang akan digunakan pada penelitian ini adalah Gray Level Co-occurrence Matrix (GLCM). *Gray Level Co-occurrence Matriks* merupakan metode tekstur statistik yang paling banyak digunakan dimana ciri tekstur diekstraksi dengan beberapa pendekatan statistik dari matriks *co-occurrence*. Dalam analisis citra, GLCM telah terbukti menjadi deskriptor tekstur yang paling kuat untuk klasifikasi data dan masalah regresi [1]. Penggunaan metode ini berdasar pada hipotesis bahwa dalam suatu tekstur akan terjadi perulangan konfigurasi atau pasangan aras keabuan. Metode GLCM termasuk dalam metode statistik dimana dalam perhitungan statistiknya menggunakan distribusi derajat keabuan (histrogram) dengan mengukur tingkat kontras, granularitas, dan kekasaran suatu daerah dari hubungan ketetanggaan antara piksel di dalam citra. Kemudian metode yang digunakan Artificial Neural Network (ANN), ANN mampu melakukan generalisasi dan ekstraksi dari suatu pola data tertentu, JST model komputasi yang terinspirasi oleh sistem saraf pusat dan digunakan dalam berbagai variasi aplikasi. ANN belajar dari berbagai pola masukan yang tersedia di dataset dan menyesuaikan koneksi bobot untuk mencapai hasil yang diharapkan [2].

Oleh karena itu, penulis mengimplementasikan sebuah sistem simulasi yang dapat membedakan daging udang lama dan daging udang baru dengan mengambil citra menggunakan metode GLCM dan ANN.

Berdasarkan uraian diatas, pada penelitian ini mengimplementasikan sebuah sistem simulasi yang dapat membedakan daging udang lama dan daging udang baru dengan mengambil citra menggunakan metode GLCM dan ANN. Maka dalam penelitian ini penulis mengambil judul **Mengidentifikasi Kualitas Udang Segar dilihat dari Warnanya Menggunakan metode Gray Level Co-occurrence Matriks (GLCM) dan Artificial Neural Network (ANN).**

1.2 Identifikasi Masalah

Berdasarkan latar belakang di atas, yang menjadi identifikasi masalah antara lain :

1. Manusia membutuhkan waktu proses yang lama untuk melakukan identifikasi.
2. Pengamatan manusia cenderung kurang teliti dalam membedakan udang segar dan udang tidak segar.

1.3 Rumusan Masalah

Dari uraian latar belakang masalah tersebut, dapat dirumuskan masalah pokok yang berkaitan yaitu :

1. Bagaimana kinerja penggunaan metode GLCM dan ANN dalam mengidentifikasi kualitas udang.
2. Bagaimana Hasil identifikasi kualitas udang menggunakan metode GLCM dan ANN.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan penelitian ini adalah :

1. Untuk mengetahui kinerja penggunaan metode GLCM dan ANN.
2. Untuk mengetahui Hasil penerapan metode GLCM dan ANN dalam mengidentifikasi kualitas udang.

1.5 Manfaat Penelitian

Manfaat Penelitian Ini Adalah Sebagai Berikut :

Mengetahui Hasil Penerapan Metode GLCM dan ANN untuk Mengidentifikasi Kualitas Udang Segar.

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Berikut merupakan tabel dari beberapa studi penelitian yang pernah dilakukan sebelumnya yang berkaitan dengan penelitian ini :

Tabel 2. 1 Tinjauan Studi

NO	PENELITI	JUDUL	TAHUN	METODE	HASIL
1.	Kusworo Adi, Neneng, R. Rizal Isnanto	Support Vector Machine Untuk Klasifikasi Citra Jenis Daging Berdasarkan Tekstur Menggunakan Ekstraksi Ciri Gray Level Co- occurrence Matrices (GLCM).	2016	<i>Gray Level Co- occurrence Matrice (GLCM)</i> dan <i>Support Vector Machine (SVM)</i>	Penelitian ini menghasilkan tingkat pengenalan terbaik yakni 87,5% berada pada jarak pengambilan 20 cm dengan jarak piksel tetangga d=2 pada arah GLCM 135 ⁰
2.	Neneng, Yusra Fernando.	Klasifikasi Jenis Daging Berdasarkan Analisa Citra Tekstur <i>Gray Level Co- occurrence Matrices (GLCM)</i> dan Warna.	2017	<i>Gray Level Co- occurrence Matrice (GLCM)</i> dan <i>Support Vector Machine (SVM)</i>	Hasil Klasifikasi terbaik yang diperoleh adalah 75,6% berada pada arah GLCM 45 ⁰ dengan jarak piksel tetangga d=3 dan arah GLCM 135 ⁰ dengan jarak piksel tetangga d=2. Nilai tersebut berada pada parameter sigma 2,1.

3.	Canny Amerilyse Caesar, Imam Cholissodin, S.Si. M.Kom, M. Ali Fauzi, S.Kom. M.Kom	Pemodelan Komposisi Pakan Kambing Peranakan Etawa (PE) untuk Optimasi Kandungan Gizi Susu Menggunakan <i>Artificial Neural Network</i> (ANN) dan <i>Particle Swarm Optimization</i> (PSO).	2018	<i>Artificial Neural Network</i> (ANN) dan <i>Particle Swarm Optimization</i> (PSO)	Dalam analisis pengujian yang dilakukan dengan kasus untuk berat badan kambing 32 kg, serta jenis pakan yang digunakan yaitu rumput Odot 70% dan rumput Raja 30% dapat menaikkan Protein hingga 3.42% dan menurunkan lemak hingga 5.19%.
----	---	--	------	---	--

2.2 Tinjauan Pustaka

2.2.1 Udang Segar

Udang merupakan salah satu hasil perikanan yang memiliki nilai ekonomis penting di Indonesia. Secara morfologi udang terdiri dari dua bagian, yakni bagian kepala yang menyatu dengan dada (*cephalothorax*) dan bagian dada (*abdomen*) yang terdapat ekor dibelakangnya.



Udang memiliki tubuh yang beruas-ruas dan seluruh tubuhnya tertutup kulit khitin yang tebal dan keras. Bagian kepala beratnya lebih kurang 36-49% dari total keeluruhan berat badan, daging 24-41% dan kulit 17-23%. Udang merupakan salah satu produk perikanan yang istimewa, memiliki aroma spesifik dan mempunyai nilai gizi cukup tinggi [3].

Pada penanganan udang segar, dilakukan penyortiran atas jenis, size dan mutu. Penyortiran mutu bertujuan untuk memisahkan udang segar dan udang yang sudah tidak layak lagi dikonsumsi, dengan menjaga kesegaran dari udang kita dapat menyimpan udang didalam freezer agar udang tidak cepat membusuk dan menghambat pertumbuhan bakteri pada udang.

2.2.2 Citra Digital

Citra digital adalah citra $f(x,y)$ dimana dilakukan diskritisasi koordinat sampling/ spasial dan diskritisasi tingkat kwantisasi (keabuan/ kecemerlangannya). Citra digital merupakan fungsi intensitas cahaya $f(x,y)$, dimana harga x dan harga y adalah koordinat spasial. Harga fungsi tersebut disetiap titik (x,y) merupakan tingkat kecemerlangan citra pada titik tersebut [4].

Citra digital merupakan suatu matriks dimana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya (yang disebut sebagai elemen gambar/ pixel/ piksel/ pels/ picture element) menyatakan tingkat keabuan pada titik tersebut. Format citra digital, matriks yang dinyatakan citra digital yaitu dengan matriks berukuran N (baris/tinggi) x M (kolom/lebar).

N = jumlah baris $0 \leq y \leq N - 1$.

M = jumlah kolom $0 \leq x \leq M - 1$.

L = maksimal warna intensitas $0 \leq f(x,y) \leq L - 1$.

(gray level/ derajat keabuan)

$$f_{(x,y)} \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

2.2.3 Computer Vision

Computer Vision adalah bidang yang mencakup metode untuk memperoleh, mengolah, menganalisis dan memahami data visual seperti gambar dan video. Tujuan utama dari *Computer Vision* adalah agar computer/mesin dapat meniru kemampuan perseptual mata manusia dan otak atau bahkan dapat mengunggulinya untuk tujuan tertentu.

Computer Vision merupakan transformasi atau perubahan dari data-data yang dapat berupa gambar diam ataupun *video* kamera menjadi bentuk lain atau satu representasi baru. *Computer vision* dapat didefinisikan dengan pengertian pengolahan citra yang dikaitkan dengan akuisisi citra, pemrosesan, klasifikasi, penganan, dan pencakupan keseluruhan, pengambilan keputusan yang diikuti pengidentifikasian citra.

Pengolahan citra adalah pengolahan suatu citra dengan menggunakan komputer, untuk menghasilkan suatu citra yang lain. Hubungan pengolahan citra dengan *computer vision* yaitu pengolahan citra merupakan proses awal (*preprocessing*) pada *computer vision*. *Computer vision* mencoba meniru cara kerja visual manusia (*Human Vision*). *Human Vision* sesungguhnya sangat kompleks, manusia melihat objek dengan indera penglihatan (mata) lalu objek citra diteruskan ke otak untuk diinterpretasi sehingga manusia mengerti objek apa yang tampak dalam pandangan matanya [5].

2.2.4 Pengenalan Pola

Pengenalan pola merupakan salah satu cabang dari ilmu kecerdasan buatan. Pengenalan pola mengelompokkan data numerik dan simbolik (termasuk citra) secara otomatis oleh mesin (komputer). Sistem pengenalan pola merupakan

komponen penting dalam proses peniruan kemampuan indra manusia terutama penglihatan dan pendengaran.

Pola adalah entitas yang terdefiniskan dan dapat diidentifikasi melalui ciri-cirinya (features). Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola lainnya. Ciri yang bagus adalah yang mempunyai daya pembeda yang tinggi, sehingga pengelompokan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan keakuratan yang tinggi. Ciri pada suatu pola diperoleh dari hasil pengukuran terhadap obyek uji. Pengenalan pola bertujuan menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Dengan kata lain, pengenalan pola membedakan suatu objek dengan objek lain [6].

2.2.5 Klasifikasi

Klasifikasi daging terbagi menjadi dua proses, yaitu *training* dan *testing* yang akan menggunakan Jaringan Syaraf Tiruan (JST) dengan metode *Backpropagation*. Tahapan Klasifikasi jenis daging yang dilakukan dalam penelitian ini meliputi citra masukan, pra pengolahan, ekstraksi ciri tekstur dan warna, ciri tekstur dan warna, menyimpan ciri tekstur dan warna ke basis data, klasifikasi menggunakan SVM, dan terakhir mendapatkan hasil klasifikasi [7].

2.2.6 Gray Level Co-occurrence Matriks

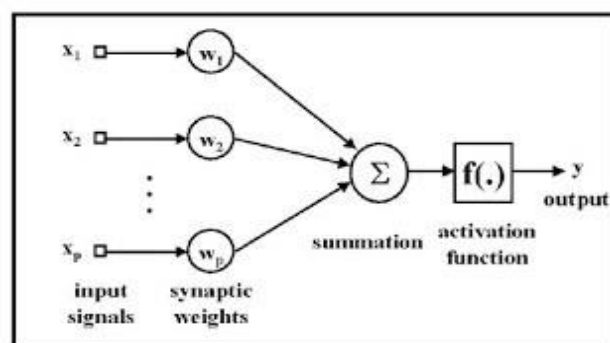
Gray Level Co-occurrence Matriks (GLCM) merupakan metode yang dapat digunakan untuk analisis tekstur statistik. GLCM telah terbukti menjadi deskriptor tekstur yang paling kuat yang digunakan dalam analisis citra. Matrik ko-okurensi dibentuk dari suatu citra dengan melihat pada piksel-piksel yang berpasangan yang memiliki intensitas tertentu. Penggunaan metode ini berdasarkan pada hipotesis bahwa dalam suatu tekstur akan terjadi perulangan konfigurasi atau pasangan aras keabuan. Misal, d didefinisikan sebagai jarak antara dua posisi piksel, yaitu (x_1, y_1) dan (x_2, y_2) , dan θ didefinisikan sebagai sudut diantara keduanya. Orientasi sudut θ terdiri atas empat arah sudut dengan interval sudut 45° , yaitu 0° , 45° , 90° , dan 135° [8].

2.2.7 Artificial Neural Network

Artificial Neural Network atau Jaringan Syaraf Tiruan (JST) adalah suatu metode komputasi yang meniru system jaringan saraf biologis, metode ini menggunakan elemen perhitungan non-linier dasar yang disebut neuron yang diorganisasikan sebagai jaringan yang saling berhubungan.

Artificial Neural Network adalah salah satu cabang ilmu dari bidang ilmu kecerdasan buatan salah satu model Jaringan Syaraf Tiruan yang sering digunakan untuk pembelajaran adalah perceptron. Metode perceptron merupakan metode pembelajaran dengan pengawasan dalam sistem jaringan syaraf. Cara kerja neural network dapat dianalogikan sebagaimana halnya manusia belajar dengan menggunakan contoh atau yang disebut supervised learning. Sebuah Neural Network dikonfigurasi untuk aplikasi tertentu, seperti pengenalan pola atau klasifikasi data, dan kemudian disempurnakan melalui proses pembelajaran [9].

No table of figures entries found.



Gambar 2.1 : Konsep Dasar Pemodelan ANN

Ket:

- Sejumlah sinyal masukan x dikalikan dengan masing-masing penimbang yang bersesuaian W
- Kemudian dilakukan penjumlahan dari seluruh hasil perkalian tersebut dan keluaran yang dihasilkan dilakukan kedalam fungsi pengaktif untuk mendapatkan tingkat derajat sinyal keluarannya $F(x.W)$
- Walaupun masih jauh dari sempurna, namun kinerja dan tiruan neuron ini indentik dengan inerja dari sel otak yang kita kenal saat ini.

- Misalkan ada n buah sinyal masukan dan n buah penimbang, fungsi keluaran dari neuron adalah seperti persamaan berikut:

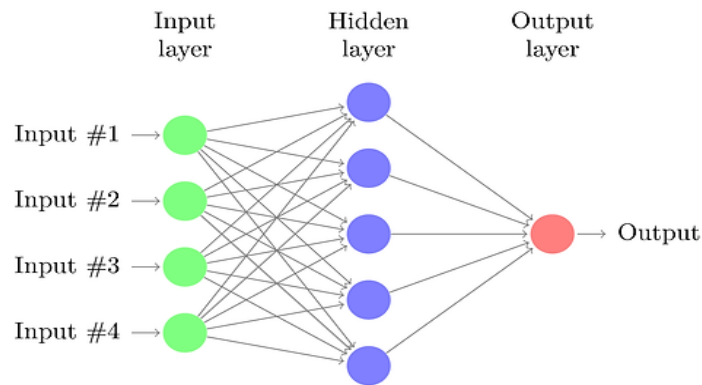
$$F(x.W) = f(w_1x_1 + \dots + w_nx_n)$$

Pada tahu 1943 Mc. Culloch dan Pitts memperkenalkan model matematika yang merupakan penyederhanaan dari struktur sel saraf yang sebenarnya.

$$y = f\left(\sum_{i=1}^n x_i w_i\right)$$

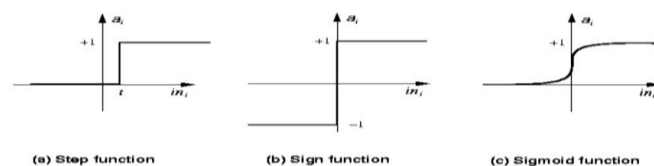
Korelasi antara ketiga komponen pada persamaan di atas yaitu: Signal x berupa vector berdimensi n ($x_1, x_2, \dots, x_n$)^T akan mengalami penguatan oleh *synapse* w ($w_1, w_2, \dots, w_n$)^T. Selanjutnya akumulasi dari penguatan tersebut akan mengalami transformasi oleh fungsi aktivasi f . Fungsi f ini akan memonitor, bila akumulasi penguatan signal itu telah melebihi batas tertentu, maka sel *neuron* yang semula berada dalam kondisi “0”, akan mengeluarkan signal “1”. Berdasarkan nilai *output* tersebut (y), sebuah *neuron* dapat berada dalam dua status: “0” atau “1”. *Neuron* disebut dalam kondisi *firing* bila menghasilkan output bernilai “1”.

Artificial Neural Network (ANN), menghasilkan model yang sulit dibaca dan dimengerti oleh manusia karena memiliki banyak layer (*Multilayer Perception*) dan sifat *non-linearity* (fungsi aktivasi). Pada bidang riset ini, ANN disebut agnostik, kita percaya, tetapi sulit membuktikan kenapa konfigurasi parameter yang dihasilkan training juga benar. konsep matematis ANN itu sendiri cukup solid, tetapi interpretability model rendah menyebabkan kita tidak dapat menganalisa proses inferensi yang terjadi pada model ANN.



Gambar 2.2 : Multilayer Perceptron

Multilayer Perception (MLP) dikenal juga sebagai *freed for-ward neural network* . istilah ANN selanjutnya mengacu pada MLP dan arsitektur lebih kompleks. *Multilayer Perception* Secara literal memiliki beberapa layers. Secara umum ada 3 layers : *input, hidden dan output layer*. Input layer menerima input (tanpa melakukan operasi apapun), kemudian nilai input (tanpa dilewatkan konfungsasi aktivasi) diberikan ke hidden units. Pada hidden units input diproses dan dilakukan perhitungan hasil fungsi aktivasi untuk tiap-tiap neuron, lalu hasilnya diberikan ke layer berikutnya. Hasil dari input layer akan diterima sebagai input bagi hidden layer. Begitu pula seterusnya hidden layer akan mengirimkan hasilnya untuk output layer.

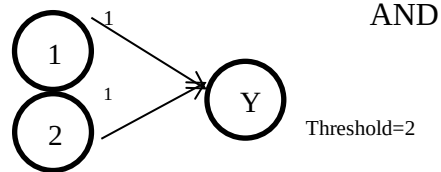


- $\text{Step}_t(x) = 1$ if $x \geq t$, else 0
- $\text{Sign}(x) = +1$ if $x \geq 0$, else -1
- $\text{Sigmoid}(x) = 1/(1+e^{-x})$
- Identity Function

Gambar 2.3 : Fungsi-fungsi Aktivasi

The First Neural Network :

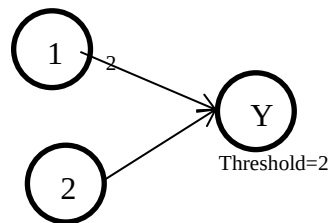
- Fungsi AND



AND

x1	x2	x3
0	0	0
0	1	0
1	0	0
1	1	1

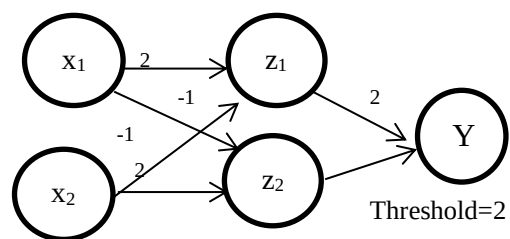
- Fungsi OR



OR

x1	x2	x3
0	0	0
0	1	1
1	0	1
1	1	1

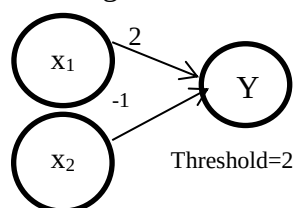
- Fungsi XOR



Threshold=2

x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	0

- Fungsi AND-NOT



AND-NOT

x1	x2	x3
0	0	0
0	1	0
1	0	1
1	1	1

2.2.8 Confusion Matrix

Pengukuran terhadap kinerja suatu system klasifikasi merupakan hal yang penting. Kinerja system klasifikasi menggambarkan seberapa baik system dalam mengklasifikasikan data. *Confusion Matix* merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi[11]

Confusion Matrix memberikan penilaian dalam performance klasifikasi berdasarkan objek dengan benar atau salah. Sebagaimana diketahui bahwa *confusion matrix* merupakan sebuah pendekatan yang sangat baik dan mudah dalam penggunaannya untuk menganalisa seberapa baik pengklasifikasian yang dilakukan untuk memperoleh 4 (empat) nilai True Positive, Falase Negaive, False Positive dan True Negative

True Positive (TP) menunjukan citra teridentifikasi secara benar sesuai dengan kelasnya (Positif). False Positive (FP) merupakan citra yang seharusnya teridentifikasi dengan tepat pada kelasnya ternyata pada proses klasifikasi namun salah dalam mengidentifikasi. True Negative (TN) merupakan citra yang bukan anggota kelas tersebut teridentifikasi tepat bukan anggota kelas tersebut (negative). False Negative (FN) menunjukan citra yang seharusnya bukan anggota dari kelas tersebut teridentifikasi sebagai anggota kelas tersebut

Nilai akurasi menggambarkan seberapa akurat system dapat mengklasifikasikan data secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data Berdasarkan nilai True Negative, True Positive, False Negatve dan False Positive dapat diperoleh nilai akurasi, presisi dan recall.

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

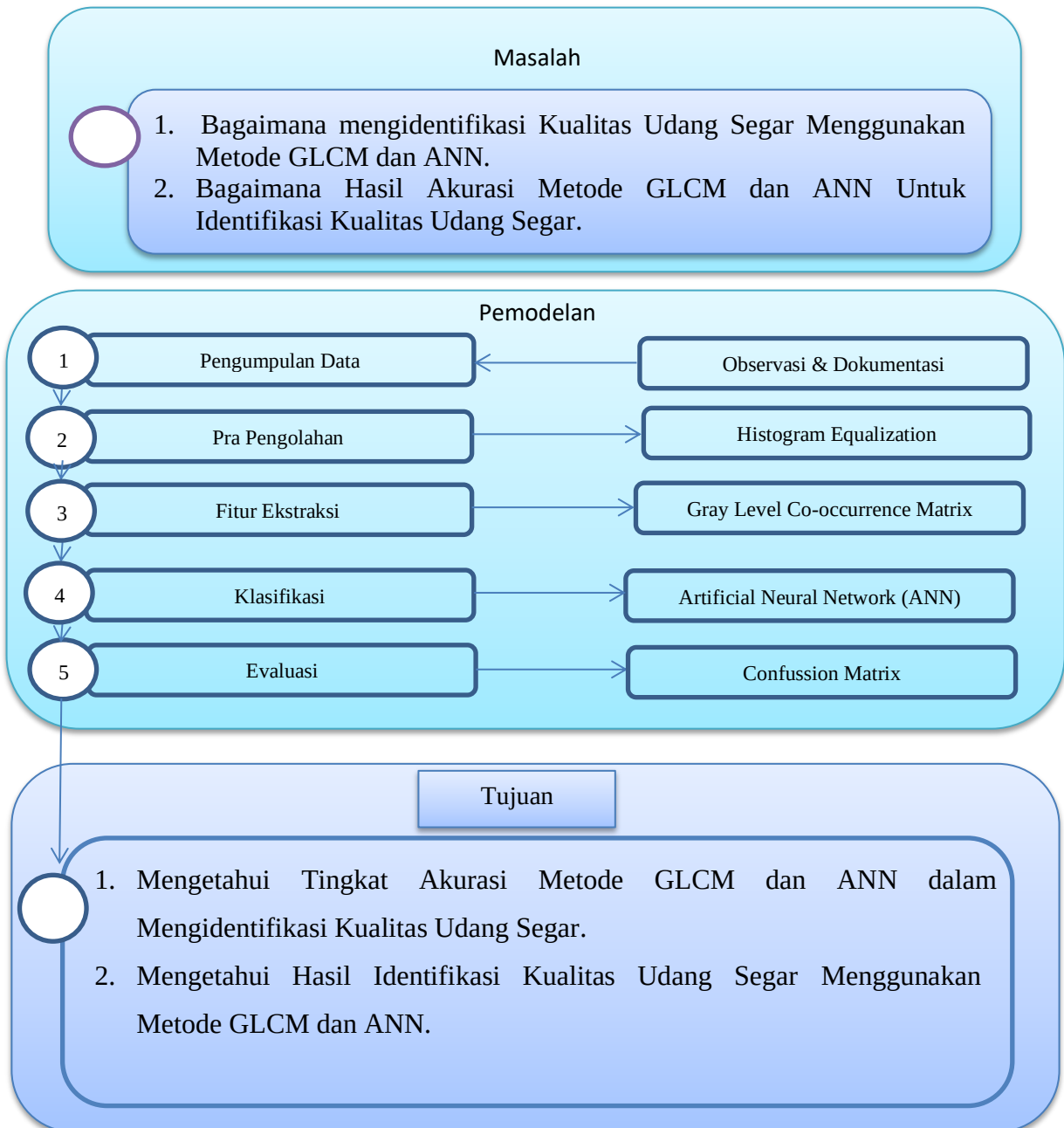
$$Presisi = \frac{TP}{TP+FP} * 100\%$$

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

Dimana :

- TP adalah True Positive yaitu jumlah data positif yang terklasifikasi dengan benar oleh system
- TN adalah True Negative yaitu jumlah data negative yang terklasifikasi dengan benar oleh system
- FN adalah False Negative yaitu jumlah data negative namun terklasifikasi salah oleh system
- FP adalah False Positive yaitu jumlah data positif namun terklasifikasi salah oleh system

2.4 Kerangka Pikir



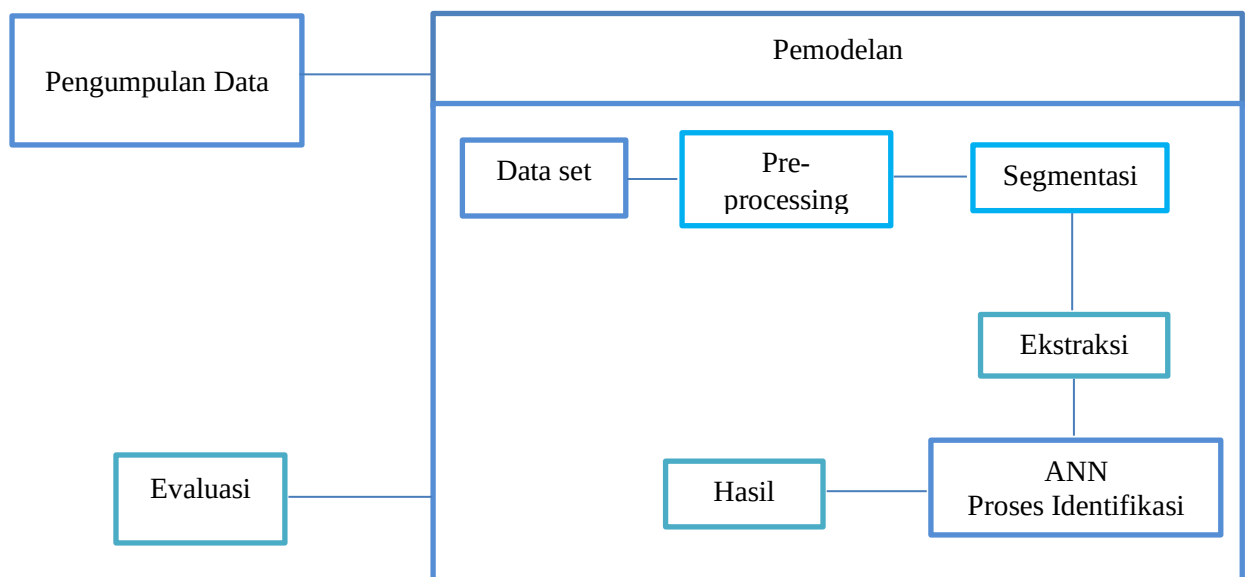
BAB III

METODE PENELITIAN

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

Penelitian ini menggunakan metode penelitian eksperimen.” Penelitian eksperimental diartikan sebagai pendekatan penelitian kuantitatif yang paling penuh, artinya memenuhi semua persyaratan untuk menguji hubungan sebab akibat. Subjek penelitian ini adalah mengidentifikasi kualitas udang segar dan udang tidak segar.

Adapun metode penelitian ini diusulkan seperti gambar dibawah ini :

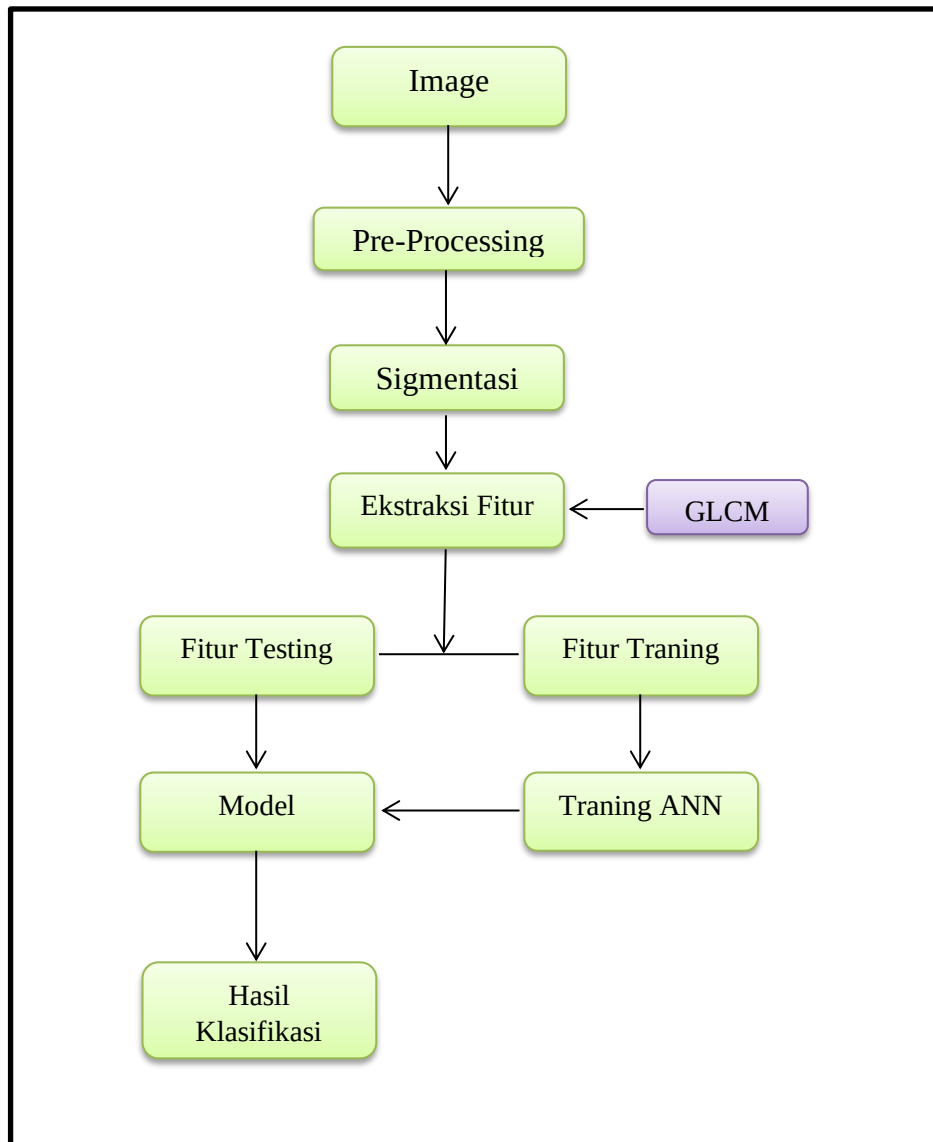


Gambar 3.1 Metode Penelitian

3.2 Pengumpulan Data

- a. Dataset pada penelitian ini adalah data langsung dari objek penelitian, maka dilakukan teknik Observasi, metode ini memungkinkan analisis sistem mengamati atau meninjau langsung. Adapun pada penelitian ini dilakukan dengan cara mengumpulkan data udang.
- b. Data literatur penelitian ini didapatkan dari beberapa referensi jurnal.

3.3 Pemodelan



3.3.1 Pre-Processing

Pre-processing dilakukan untuk menghasilkan ciri atau fitur penting dari sebuah citra. Pada penelitian ini *Pre-processing* menggunakan *Histogram Equalization* yang berfungsi untuk mendistribusi nilai derajat keabuan pada suatu citra dibuat rata.

3.3.2 Ekstraksi Fitur Warna

Ekstraksi fitur berfungsi sebagai pendeteksi ciri dari suatu citra. Ciri yang dapat digunakan untuk membedakan objek satu dengan objek yang lainnya, diantaranya ciri bentuk, ciri ukuran, ciri geometri, ciri tekstur dan ciri warna. Pada penelitian ini digunakan ekstraksi ciri menggunakan *Gray Level Co-occurrence Matrix*. Masing-masing objek diekstrak cirinya berdasarkan parameter-parameter tertentu dan dikelompokkan pada kelas tertentu. Nilai dan parameter-parameter tersebut kemudian dijadikan sebagai data masukan dalam proses identifikasi/klasifikasi.

3.3.3 Fitur Training

Setelah fitur citra diekstrak menggunakan GLCM selanjutnya akan di training menggunakan Algoritma *Artificial Neural Network*.

3.3.4 Fitur Testing

Fitur testing yang dihasilkan kemudian diklasifikasikan menggunakan algoritma *Artificial Neural Network* untuk mendapatkan hasil/bobot terbaik. Fitur testing digunakan untuk mengukur sejauh mana *classifier* berhasil melakukan klasifikasi.

3.3.5 Model

Model merupakan hasil dari proses training dari algoritma *Artificial Neural Network* menggunakan data training.

3.4 Evaluasi Model

Model yang telah dihasilkan kemudian dievaluasi dengan menggunakan *Confusion Matrix* untuk mengetahui *Akurasi*. Evaluasi model dalam penelitian ini dilakukan untuk menguji performansi dari sistem yang dibangun.

BAB IV

HASIL PENELITIAN

4.1 Hasil pengumpulan data

Data yang digunakan dalam penelitian ini adalah data yang dikumpulkan sendiri dengan mengambil gambar menggunakan kamera handphone dengan ukuran bervariasi.



Gambar 4.1. Hasil pengumpulan data

Gambar diatas merupakan hasil pengumpulan dataset yang telah dilakukan dalam penelitian ini, dimana dataset yang digunakan berupa gambar udang yang secara detail dijelaskan sebagai berikut :

- a. Objek : udang
- b. Jenis data : Gambar
- c. Format File : JPEG
- d. Ukuran : bervariasi
- e. Jumlah Kelas : 2 (Segar dan Tidak Segar)
- f. Segar : 81 gambar

- g. Tidak Segar : 61 gambar
- h. Jumlah Objek : 26 objek
- i. Total Data : 142 gambar
- j. Analisis dan Pemodelan
- k. Pra pengolahan data
- l. Konversi gambar dari RGB ke Grayscale

Pada tahapan pra pengolahan yang dilakukan pada penelitian ini adalah dengan melakukan proses konversi gambar RGB yang diperoleh dari proses pengumpulan data gambar menjadi bentuk gray scale atau skala abu-abu citra, proses konversi gambar menjadi gray scale secara matematis dapat dilakukan dengan mencari nilai rata-rata untuk setiap channel gambar sehingga gambar yang dihasilkan nanti hanya tinggal 1 channel saja (1 layer).

$$g(x) = \frac{R + G + B}{3}$$

120	122	134	210	90	83
127	166	112	147	45	42
120	156	79	94	23	21
99	125	51	57	11	10
75	91	31	34	6	5
53	63	18	20	3	3

Channel R

105	107	119	195	75	68
112	151	97	132	30	27
105	141	64	79	38	36
84	110	36	42	26	25
60	76	16	19	21	20
38	48	3	5	18	18

Channel G

130	132	144	220	100	93
137	176	122	157	55	52
130	166	89	104	63	61
109	135	61	67	51	50
85	101	41	44	46	45
63	73	28	30	43	43

Channel B

118	120	132	208	88	81
125	164	110	145	43	40
118	155	77	92	41	39
97	123	49	56	30	29
73	90	29	32	24	24
51	61	17	18	21	21

Gray Scale

4.2 Peningkatan kualitas citra

Pada tahapan berikutnya pada proses pra pengolahan citra yang dilakukan pada penelitian ini adalah dengan melakukan proses peningkatan kualitas citra guna mendapatkan gambar atau citra yang memadai untuk proses lebih lanjut, proses ini perlu dilakukan karena pada saat dilakukan proses pengambilan gambar terdapat sejumlah gambar yang memiliki kualitas citra yang kurang memadai seperti gambar terlalu gelap atau terlalu terang, sehingga perlu dilakukan proses peningkatan citra guna menyeragamkan kondisi citra. Proses peningkatan kualitas citra sendiri pada penelitian ini dilakukan dengan cara operasi morfologi spasial pada setiap pixel citra dengan menggunakan kernel berukuran 3 x 3 dengan komposisi filter sebagai berikut :

0	-1	0
-1	5	-1
0	-1	0

Gambar 4.2 Filter 3 x 3

Proses peningkatan kualitas citra dilakukan dengan melakukan operasi morfologi citra terhadap gambar gray scale citra yang telah diperoleh dari tahap sebelumnya dengan memanfaatkan filter 3 x 3 diatas, secara matematis proses morfologi tersebut dapat dijelaskan sebagai berikut :

118	120	132	208	88	81
125	164	110	145	43	40
118	155	77	92	41	39
97	123	49	56	30	29
73	90	29	32	24	24
51	61	17	18	21	21

Gambar 4.3 Pixel Citra

0	-1	0
-1	5	-1
0	-1	0

Gambar 4.4 kernel/filter yang digunakan

Secara kerja proses morfologi dalam melakukan proses peningkatan kualitas citra adalah dengan melakukan proses perkalian antara kernel/filter yang ada dengan nilai pixel citra yang seukuran dengan ukuran filter yang digunakan yakni 3x3, proses perkalian tersebut dilakukan secara linier seperti yang dicontoh sebagai berikut :

$$g(x) = (0 * 118 + -1 * 120 + 0 * 132 + -1 * 125 + 5 * 164 + -1 * 110 + 0 * 118 + -1 * 155 + 0 * 77) / 9$$

$$g(x) = 35$$

sehingga hasil $g(x)$ tersebut dimasukkan kedalam matriks citra baru seperti ditampilkan pada gambar berikut :

118	120	132	208	88	81
125	35				40
118					39
97					29
73					24
51	61	17	18	21	21

Gambar 4.5 Matriks hasil proses morfologi

Proses tersebut akan terus berulang hingga seluruh matriks tersebut terisi secara penuh. Seperti di bawah ini

118	120	132	208	88	81
125	35	37	30	11	40
118	32	40	29	18	39
97	42	26	34	12	29
73	45	45	23	13	24
51	61	17	18	21	21

Gambar 4.6 hasil akhir proses morfologi

4.3 Segmentasi

Proses berikutnya dalam tahapan pra pengolahan data gambar adalah melakukan proses segmentasi citra yang bertujuan untuk memisahkan gambar objek yang diteliti dalam hal ini gambar udang dengan bagian background atau gambar latar belakang dari suatu citra. Adapun pendekatan yang digunakan dalam proses segmentasi adalah dengan cara melakukan proses thresholding pada citra hasil proses peningkatan kualitas gambar. dalam pendekatan proses thresholding salah satu parameter yang paling penting dalam proses tersebut adalah penentuan nilai threshold yang akan digunakan guna mendapatkan hasil segmentasi terbaik. Dalam penelitian ini penentuan nilai threshold yang digunakan adalah dengan menggunakan metode OTSU.

4.4 Ekstraksi fitur menggunakan GLCM

Tahapan berikutnya dalam penelitian ini adalah dengan melakukan proses ekstraksi fitur citra. Adapun citra yang di proses pada tahapan ini adalah citra hasil proses segmentasi yang telah diperoleh dari tahapan sebelumnya. Metode ekstraksi fitur yang digunakan dalam penelitian ini adalah metode gray level co-occurrence matrix (GLCM). Berdasarkan penjelasan secara teori sebagaimana yang telah dijelaskan pada landasan teori pada bab 2, metode GLCM memiliki 2 parameter yang digunakan yakni nilai angle yang memiliki 4 buah nilai pilihan yang dapat digunakan yakni 0, 45, 90 dan 135 yang dinyatakan dalam bentuk derajat (arah) dan parameter kedua adalah nilai ketetanggaan yang menyatakan jarak antar nilai antar pixel yang akan digunakan, nilai ini dapat berupa angka 1, 2, 3, 4 dan seterusnya. Adapun nilai angle yang digunakan dalam penelitian ini adalah nilai 0 derajat dengan jarak antar pixel adalah 1

Cara kerja dari ekstraksi fitur menggunakan metode GLCM secara matematis dapat dijelaskan sebagai berikut :

Langkah ke-1

Menyiapkan gambar yang akan di ekstraksi kemudian mengubah citra tersebut ke bentuk skala abu-abu.

118	120	132	208	88	...	187	187	165	210	223
125	35	37	30	11	...	128	190	165	209	223
118	32	40	29	18	...	128	200	154	221	223
97	42	26	34	12	...	170	203	165	210	223
73	45	45	23	13	...	209	210	165	223	223
...	...	...	...	...	...	...	...	...	...	...
128	200	154	118	120	...	125	35	37	30	11
170	203	165	125	35	...	118	32	40	29	18
165	210	154	221	223	...	97	42	26	34	12
165	209	165	210	223	...	73	45	45	23	13
154	221	165	223	223	...	32	40	29	45	23

Gambar 4.7 nilai pixel citra hasil proses segmentasi

Langkah ke-2

Menetapkan parameter GLCM yang digunakan yakni parameter angle atau arah orientasi pembentukan matrix glcm dan jarak ketetanggaan nilai pixel, dalam penelitian ini akan digunakan nilai angle = 0^0 (0 derajat) dan jarak ketetanggaan sebesar 1 jarak pixel

Langkah ke-3

Langkah berikutnya adalah membentuk matrix glcm berdasarkan parameter yang digunakan terhadap citra hasil proses segmentasi yang telah di ubah ke skala abu-abu.

	0	1	2	3	...	221	222	223	224	225
0										
1										
2										
3										
...										
221										
222										
223										
224										
225										

Gambar 4.8 Matrix glcm yang belum terisi

Kemudian untuk proses pengisian matrix tersebut diatas akan dilakukan pengecekan terhadap setiap nilai pixel pada citra yang ada berdasarkan parameter input yang digunakan yakni nilai angle sebesar 0^0 dan jarak antar pixel sebesar 1 poin, sehingga matrix glcm yang diperoleh adalah sebagai berikut :

	0	1	2	3	...	251	252	253	254	255
0	0	0	0	0	...	0	0	0	0	0
1	0	0	0	0	...	0	0	0	0	0
2	0	0	0	0	...	0	0	0	0	0
3	0	0	0	0	...	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
251	0	0	0	0	...	2	1	2	0	2
252	0	0	0	0	...	1	2	2	2	5
253	0	0	0	0	...	2	2	2	3	13
254	0	0	0	0	...	0	2	3	6	32
255	0	0	0	0	...	2	5	13	32	378

Gambar 4.9 Hasil proses pencarian matrix glcm

Langkah ke-4

Langkah selanjutnya dalam proses ekstraksi fitur adalah dengan menghitung informasi nilai statistik dari matrix glcm yang telah diperoleh pada proses sebelumnya, adapun informasi nilai statistik yang digunakan adalah nilai contrast, energy, homogeneity, correlation, dan angular second moment (ASM) yang mana rumus persamaan nilai statistik tersebut telah dicantumkan dalam

bab 2. Adapun hasil ekstraksi fitur dari seluruh data training adalah sebagai berikut :

Tabel 4.1 Hasil proses ekstraksi fitur pada data training

No	ASM	Contrast	Energy	Homogeneity	Correlation
1	0.36	564.29	0.60	0.70	0.95
2	0.61	800.86	0.78	0.81	0.89
3	0.33	558.38	0.58	0.68	0.95
4	0.43	711.82	0.66	0.70	0.90
5	0.67	290.31	0.82	0.86	0.91
6	0.46	362.74	0.68	0.75	0.93
7	0.47	353.75	0.68	0.75	0.93
8	0.53	409.74	0.73	0.79	0.91
9	0.45	550.56	0.67	0.73	0.88
10	0.44	514.29	0.66	0.72	0.89
11	0.47	521.56	0.69	0.74	0.88
12	0.62	243.99	0.78	0.84	0.93
13	0.58	315.47	0.76	0.83	0.93
14	0.56	318.05	0.75	0.82	0.93
15	0.33	557.32	0.58	0.67	0.92
16	0.35	518.06	0.60	0.69	0.92
17	0.38	424.75	0.62	0.71	0.93
18	0.05	412.03	0.22	0.40	0.95
19	0.08	385.24	0.28	0.45	0.96
20	0.07	349.55	0.26	0.43	0.96
21	0.09	393.00	0.31	0.47	0.96
22	0.27	422.94	0.52	0.61	0.96
23	0.30	387.47	0.55	0.63	0.97
24	0.08	640.22	0.29	0.44	0.93
25	0.06	751.03	0.24	0.39	0.92
26	0.07	666.32	0.27	0.42	0.93
27	0.06	776.20	0.24	0.38	0.92
28	0.10	794.36	0.32	0.45	0.92
29	0.12	754.93	0.34	0.48	0.93
30	0.69	251.22	0.83	0.87	0.94
31	0.47	436.14	0.68	0.76	0.93
32	0.58	360.70	0.76	0.82	0.93
33	0.58	582.93	0.76	0.80	0.88

No	ASM	Contrast	Energy	Homogeneity	Correlation
34	0.53	622.70	0.73	0.77	0.88
35	0.56	575.25	0.75	0.79	0.88
36	0.50	575.54	0.71	0.76	0.89
37	0.57	298.19	0.75	0.82	0.93
38	0.55	322.76	0.74	0.81	0.93
39	0.19	362.33	0.43	0.54	0.97
40	0.27	402.94	0.52	0.61	0.97
41	0.28	430.80	0.53	0.61	0.97
42	0.25	429.81	0.50	0.59	0.97
43	0.15	357.54	0.38	0.55	0.97
44	0.36	553.83	0.60	0.66	0.96
45	0.26	344.26	0.51	0.61	0.97
46	0.27	333.75	0.52	0.62	0.97
47	0.24	311.12	0.49	0.59	0.97
48	0.10	530.07	0.32	0.50	0.94
49	0.04	425.61	0.21	0.42	0.94
50	0.04	480.82	0.21	0.41	0.93
51	0.04	449.18	0.19	0.41	0.94
52	0.05	503.75	0.23	0.43	0.93
53	0.00	323.97	0.02	0.17	0.79
54	0.00	343.30	0.02	0.16	0.79
55	0.00	344.64	0.02	0.17	0.81
56	0.17	467.14	0.41	0.55	0.94
57	0.18	461.92	0.42	0.54	0.94
58	0.16	451.10	0.41	0.53	0.94
59	0.16	469.28	0.40	0.52	0.94
60	0.16	432.24	0.41	0.53	0.94
61	0.44	312.17	0.66	0.71	0.96
62	0.43	330.38	0.66	0.71	0.96
63	0.40	353.87	0.63	0.69	0.96
64	0.51	425.62	0.71	0.78	0.92
65	0.51	378.62	0.71	0.78	0.93
66	0.52	372.91	0.72	0.78	0.93
67	0.57	555.48	0.75	0.81	0.91
68	0.58	506.56	0.76	0.81	0.91
69	0.56	571.84	0.75	0.80	0.90
70	0.56	497.30	0.75	0.80	0.92
71	0.52	352.71	0.72	0.79	0.93

No	ASM	Contrast	Energy	Homogeneity	Correlation
72	0.50	390.00	0.71	0.77	0.92
73	0.49	415.59	0.70	0.77	0.91
74	0.24	461.43	0.49	0.64	0.96
75	0.20	620.31	0.44	0.59	0.95
76	0.23	543.11	0.48	0.62	0.96
77	0.24	567.49	0.49	0.64	0.96
78	0.18	705.65	0.42	0.55	0.94
79	0.18	678.45	0.43	0.57	0.94
80	0.18	660.12	0.42	0.58	0.95
81	0.16	729.25	0.40	0.55	0.94
82	0.30	603.73	0.55	0.65	0.94
83	0.55	929.88	0.74	0.78	0.88
84	0.28	690.39	0.53	0.64	0.94
85	0.38	787.17	0.62	0.68	0.92
86	0.30	603.52	0.55	0.67	0.95
87	0.38	811.66	0.62	0.70	0.91
88	0.37	784.67	0.61	0.74	0.94
89	0.46	635.49	0.68	0.74	0.92
90	0.00	460.92	0.02	0.11	0.85
91	0.00	450.47	0.05	0.16	0.90
92	0.00	397.50	0.03	0.15	0.89
93	0.00	374.48	0.02	0.13	0.86
94	0.00	289.66	0.03	0.17	0.88
95	0.00	379.97	0.02	0.14	0.84
96	0.00	395.03	0.02	0.12	0.81
97	0.00	409.22	0.02	0.13	0.84
98	0.00	292.56	0.02	0.14	0.84
99	0.20	912.93	0.45	0.59	0.93
100	0.56	473.91	0.75	0.78	0.88
101	0.35	603.65	0.59	0.67	0.88
102	0.30	636.78	0.54	0.63	0.89
103	0.36	635.70	0.60	0.67	0.88
104	0.35	601.47	0.59	0.66	0.88
105	0.48	551.86	0.69	0.74	0.89
106	0.51	505.64	0.72	0.76	0.88
107	0.45	518.59	0.67	0.73	0.89
108	0.38	308.39	0.62	0.70	0.93
109	0.50	380.17	0.71	0.76	0.91

No	ASM	Contrast	Energy	Homogeneity	Correlation
110	0.44	318.32	0.66	0.74	0.93
111	0.44	284.76	0.66	0.73	0.94
112	0.43	314.49	0.66	0.73	0.94
113	0.38	401.63	0.62	0.68	0.91
114	0.36	405.27	0.60	0.67	0.91
115	0.36	382.93	0.60	0.67	0.92
116	0.33	509.12	0.58	0.66	0.91
117	0.30	424.52	0.54	0.68	0.93
118	0.36	354.81	0.60	0.72	0.96
119	0.34	441.95	0.58	0.70	0.95
120	0.49	254.60	0.70	0.79	0.96
121	0.43	325.71	0.66	0.76	0.96
122	0.35	605.67	0.59	0.71	0.94
123	0.36	574.70	0.60	0.71	0.94
124	0.36	553.08	0.60	0.71	0.94
125	0.36	593.16	0.60	0.72	0.94
126	0.30	545.55	0.55	0.68	0.95
127	0.34	488.20	0.58	0.71	0.95
128	0.35	572.56	0.59	0.72	0.94
129	0.36	502.83	0.60	0.72	0.95
130	0.34	530.53	0.58	0.72	0.95
131	0.34	541.45	0.59	0.72	0.95
132	0.35	724.21	0.59	0.72	0.94
133	0.35	659.56	0.59	0.71	0.94
134	0.38	399.10	0.61	0.75	0.96
135	0.33	511.64	0.57	0.72	0.95
136	0.18	784.58	0.43	0.59	0.94
137	0.24	651.38	0.49	0.69	0.95
138	0.28	787.34	0.53	0.71	0.94
139	0.25	854.33	0.50	0.69	0.93
140	0.24	792.41	0.49	0.69	0.94
141	0.19	946.11	0.44	0.67	0.93
142	0.28	699.08	0.53	0.71	0.94

4.5 Proses klasifikasi menggunakan ANN

Tahapan selanjutnya dalam penelitian ini adalah melakukan proses klasifikasi, dimana proses klasifikasi akan dilakukan dengan menggunakan algoritma Artificial Neural Network (ANN). Dalam melakukan proses klasifikasi menggunakan algoritma ANN diperlukan beberapa langkah sebelum memperoleh hasil akhir berupa kelas atau kategori dari gambar testing yang diproses. Adapun langkah-langkah dalam proses klasifikasi menggunakan algoritma ANN adalah sebagai berikut :

4.5.1 Proses transformasi data

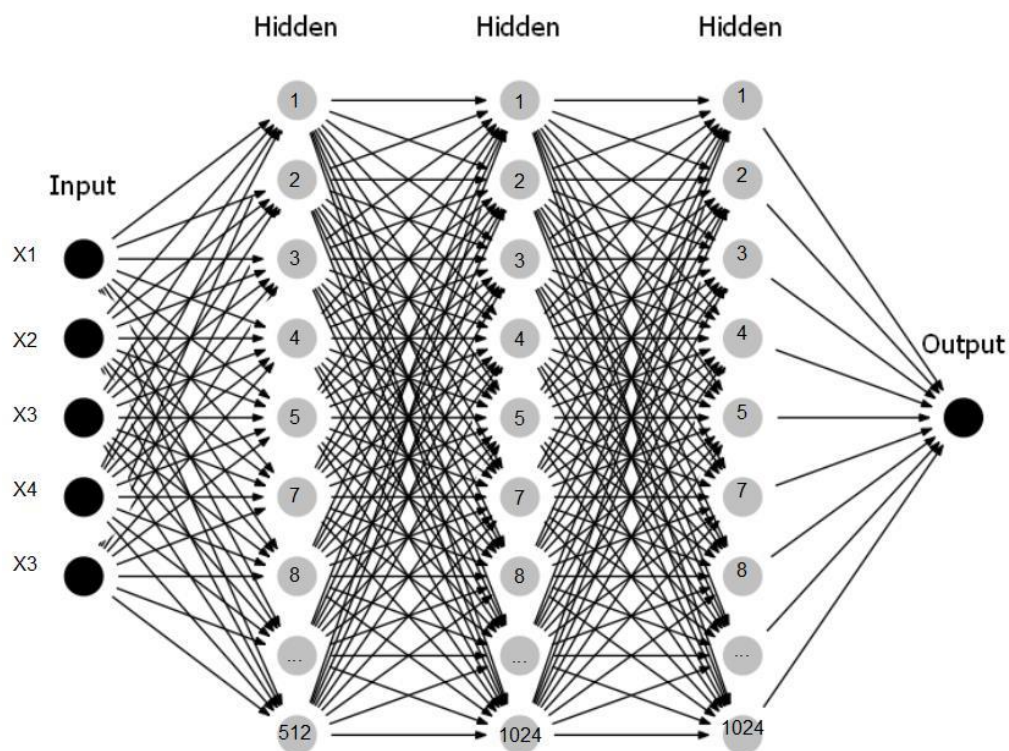
Karena algoritma yang digunakan adalah algoritma ANN, dimana data yang diolah adalah data berbentuk numerik sehingga data yang diolah tersebut perlu untuk di normalisasi. Proses normalisasi data yang digunakan dalam penelitian ini adalah penskalaan standar (standard scaler). Hal ini perlu dilakukan agar setiap data yang diolah berada dalam skala yang sama sehingga akan lebih mengoptimalkan algoritma ANN dalam menemukan bobot terbaik untuk setiap data gambar yang diproses. Berikut ini hasil proses normalisasi data menggunakan standard scaler.

Tabel 4.2 Hasil transformasi data training

No	X1	X2	X3	X4	X5
0	1.348967	-0.0426	1.075845	0.96543	-0.15776
1	1.631831	1.874981	1.234514	1.001288	-1.03284
2	-1.30711	-0.75047	-1.12748	-0.96748	1.008961
3	-1.75835	-1.3542	-2.36618	-2.51476	-1.23389
4	1.399544	-0.19036	1.104729	0.858133	-1.27712
5	-0.055	0.262217	0.150912	0.28626	0.628687
6	1.216738	-0.59571	0.9992	0.901013	-0.35889
7	0.899495	0.110679	0.808065	0.637141	-1.11969
...	...	...	...	...	...
111	0.143282	0.167335	0.300179	0.495061	0.659575
112	0.244232	0.446363	0.37319	0.466062	0.465562

4.5.2 Training data

Setelah dilakukan proses transformasi data, langkah selanjutnya adalah melakukan proses traning data menggunakan algoritma ANN, proses training data ini bertujuan untuk menemukan bobot terbaik untuk setiap neuron sebagaimana digambarkan dalam arsitektur ANN yang disajikan di berikut ini.



Gambar 4.10 Arsitektur ANN.

Arsitektur Artificial Neural Network yang digunakan pada penelitian ini

Berdasarkan data hasil ekstraksi fitur, dimana setiap gambar diwakili oleh 5 nilai fitur statistik sebagaimana dijelaskan sebelumnya pada bagian ekstraksi fitur, sehingganya pada proses training data terdapat 5 inputan nilai yakni X1, X2, X3, X4, dan X5. Seperti yang disajikan pada gambar diatas, algoritma ANN yang digunakan memiliki arsitektur yang terdiri atas 3 hidden layer, yang setiap hidden layer terdiri dari sejumlah neuron yang masing terdiri atas : hidden layer ke-1 memiliki 512 neuron, hidden layer ke-2 memiliki 1024 neuron dan hidden layer

ke-3 memiliki 1024 neuron, kemudian hanya memiliki satu unit output karena kelas data yang dimiliki hanya dua (2) atau binary class. Untuk proses training data, secara matematis proses training untuk menemukan bobot terbaik pada algoritma ANN dijelaskan sebagai berikut :

1. Bobot awal untuk neuron 1 pada hidden layer 1 (HL1.1) = [0.1, 0.1, -0.2, 0.2, -0.1]
2. Bobot awal untuk neuron 2 pada hidden layer 1 (HL1.2) = [0.1, 0.2, 0.2, 0.1, -0.3]
3. Learning Rate : 0.001
4. Jumlah epochs : 400
5. Data training ke-1 :

Tabel 4.3 Data training ke-1

X1	X2	X3	X4	X5
1.348967	-0.0426	1.075845	0.96543	-0.15776

$$HL1.1 = 1.348967 * 0.1 + -0.0426 * 0.1 + 1.075845 * -0.2 + 0.96543 * 0.2 + -0.15776 * -0.1$$

$$= 0.1243$$

$$HL1.2 = 1.348967 * 0.1 + -0.0426 * 0.2 + 1.075845 * 0.2 + 0.96543 * 0.1 + -0.15776 * -0.3$$

$$= 0.485$$

Seluruh neuron yang terdapat di hidden layer ke-1 (hingga neuron ke 512) dihitung dengan cara yang sama dengan diatas tentunya dengan nilai bobot awal yang ditentukan secara random (acak). Selanjutnya proses perhitungan akan dilanjutkan ke hidden layer ke-2 mulai dari neuron 1 hingga neuron ke 1024, pada hidden layer ke-2 yang menjadi nilai input adalah hasil output dari setiap neuron

yang terdapat pada hidden layer ke-1, demikian pula dengan hidden layer ke-3, cara yang sama juga dilakukan pada setiap neuron dimana hasil output dari neuron yang terdapat pada hidden layer ke-2. Demikian seterusnya hingga pada output layer, apabila pada output layer (aktual output) memiliki hasil output yang berbeda dengan kelas target yang terdapat pada data training, maka nilai bobot awal tersebut akan diperbaharui dengan menambah nilai bobot awal tersebut sebesar nilai learning rate yang ditentukan (0.001). proses ini akan terus berlanjut hingga ditemukan nilai bobot terbaik, nilai bobot sudah dinyatakan terbaik apabila nilai error atau selisih antara aktual output dengan kelas target sudah memenuhi nilai minimum error yang ditentukan misalnya 0.01. berikut perolehan nilai bobot terbaik yang diperoleh dari proses training yang dilakukan :

Tabel 4.4 Nilai bobot Hidden Layer 1

Input	HL1.1	HL1.2	HL1.3	HL1.4	HL1.5	...	HL1.512
1	0.008	0.042	0.082	-0.107	-0.003	...	-0.069
2	0.076	0.023	-0.069	0.057	-0.038	...	0.100
3	-0.005	0.013	0.054	0.049	-0.014	...	-0.056
4	-0.075	0.087	-0.061	-0.045	-0.001	...	0.047
5	-0.003	-0.022	-0.109	0.002	0.011	...	0.040

Tabel diatas menampilkan nilai bobot hasil training yang dilakukan untuk hidden layer 1 yang terdiri atas 5 input dan 512 neuron.

Tabel 4.5 Nilai bobot Hidden Layer 2

Input	HL2.1	HL2.2	HL2.3	HL2.4	HL2.5	...	HL2.1024
1	0.048	0.018	0.052	-0.042	0.018	...	-0.027
2	0.025	-0.050	0.022	-0.057	0.033	...	0.001
3	0.036	-0.040	0.044	-0.025	-0.031	...	-0.054
4	-0.052	-0.037	0.031	0.038	0.018	...	-0.021
5	0.030	-0.022	-0.044	0.003	0.042	...	0.004
6	-0.010	0.058	-0.018	0.028	-0.036	...	0.014
7	-0.054	0.003	0.036	0.027	0.021	...	0.025
8	0.038	0.010	-0.061	0.001	-0.022	...	0.025
9	-0.006	0.054	-0.038	0.004	0.007	...	0.018
10	-0.022	-0.040	-0.022	0.019	0.052	...	-0.013
...	...	...	...	...	...	...	...
512	0.057	-0.014	-0.037	-0.043	-0.022	...	0.044

Tabel diatas menunjukkan nilai bobot hasil training data yang dilakukan untuk hidden layer 2 yang terdiri atas 1024 neuron dan 512 nilai input yang bersumber dari output hidden layer 1.

Tabel 4.6 Nilai bobot hidden layer 3

Input	HL3.1	HL3.2	HL3.3	HL3.4	HL3.5	...	HL3.1024
1	0.035	-0.021	-0.038	0.031	0.032	...	0.004
2	0.039	-0.028	-0.041	0.027	0.001	...	-0.034
3	0.041	0.027	-0.028	0.053	-0.049	...	-0.021
4	-0.038	0.019	-0.054	-0.012	0.032	...	0.019
5	-0.012	0.025	0.024	0.039	0.026	...	0.009
6	-0.005	-0.019	0.007	0.027	0.016	...	-0.032
7	-0.036	-0.021	-0.024	0.029	-0.001	...	-0.028
8	-0.028	0.047	0.001	0.004	-0.038	...	-0.007

Input	HL3.1	HL3.2	HL3.3	HL3.4	HL3.5	...	HL3.1024
9	-0.017	0.014	0.039	-0.040	0.022	...	0.047
10	-0.053	-0.044	0.037	-0.044	-0.017	...	0.010
...	...	...	...	...	...	...	...
1024	0.033	-0.038	0.045	-0.019	-0.036	...	0.036

Tabel diatas menunjukkan nilai bobot hasil training data yang dilakukan untuk hidden layer 3, dimana hidden layer 3 ini terdiri atas 1024 unit neuron dan 1024 nilai input yang bersumber dari output pada hidden layer 2.

Tabel 4.7 Nilai bobot Output Layer

Input	1	2	3	4	5	...	1024
Bobot	-0.016	0.011	-0.022	-0.003	-0.017	...	-0.022

Tabel diatas menunjukkan nilai bobot hasil training data untuk Output layer, dimana output layer hanya terdiri dari 1 (satu) unit neuron karena jumlah kelas pada dataset yang digunakan hanya 2 (dua) kelas saja, atau biasa disebut *binary classification*. Sementara untuk jumlah unit untuk nilai input terdiri atas 1024 unit nilai, yang mana nilai input tersebut bersumber dari output pada hidden layer 3.

4.6 Hasil klasifikasi

Proses klasifikasi dilakukan terhadap data testing yang ada, dimana data testing ini diambil dari dataset yang ada kemudian di bagi menjadi dua bagian yakni data training dan data testing. Adapun porsi pembagian data testing sebesar 20% dari dataset.

Untuk proses penentuan hasil klasifikasi atau kelas dari masing-masing data testing dilakukan secara matematis sama dengan proses training data, hanya saja bobot yang digunakan pada proses ini sudah menggunakan seluruh bobot terbaik yang diperoleh dari training data.

Tabel 4.8 Hasil proses klasifikasi menggunakan algoritma ANN

NO	ASM	Contrast	Energy	Homogeneity	Correlation	Kelas Target	Kelas Prediksi
1	1.482	-1.191	1.151	1.103	0.117	Segar	Segar
2	1.074	-0.792	0.914	0.829	0.145	Segar	Segar
3	-0.354	-0.469	-0.091	-0.212	1.197	Segar	Segar
4	1.688	-1.643	1.265	1.147	0.131	Segar	Segar
5	0.796	-0.893	0.744	0.672	0.028	Segar	Segar
6	-1.76	-0.599	-2.393	-2.693	-2.338	Tidak Segar	Tidak Segar
7	0.67	-1.197	0.663	0.57	0.311	Tidak Segar	Tidak Segar
8	0.136	-0.392	0.295	0.388	0.762	Tidak Segar	Tidak Segar
9	-1.373	-0.976	-1.23	-1.072	1.115	Segar	Segar
10	0.095	0.343	0.264	0.303	0.618	Segar	Segar
11	-1.343	1.025	-1.183	-1.128	0.085	Segar	Segar
12	0.804	0.83	0.748	0.644	-0.077	Tidak Segar	Segar
13	1.372	0.45	1.089	0.921	-1.215	Segar	Segar
14	2.12	-1.597	1.493	1.354	0.367	Segar	Segar
15	1	-0.559	0.87	0.781	-0.311	Segar	Segar
16	0.064	0.048	0.241	0.526	0.779	Tidak Segar	Tidak Segar
17	-0.197	1.177	0.039	0.084	0.318	Tidak Segar	Tidak Segar
18	0.661	-1.126	0.657	0.759	1.015	Tidak Segar	Tidak Segar
19	1.042	0.452	0.895	0.755	-1.009	Segar	Segar
20	0.391	-0.501	0.476	0.437	0.244	Segar	Segar
21	-1.238	-0.701	-1.028	-0.859	1.054	Segar	Segar
22	-0.102	0.838	0.114	0.002	-0.855	Tidak Segar	Tidak Segar
23	-1.452	1.719	-1.366	-1.351	-0.221	Segar	Segar
24	-0.085	0.628	0.128	0.221	0.679	Tidak Segar	Segar
25	-1.761	-0.818	-2.409	-2.725	-1.667	Tidak Segar	Tidak Segar
26	-1.176	0.164	-0.943	-0.665	0.36	Segar	Segar
27	0.163	0.236	0.315	0.54	0.734	Tidak Segar	Tidak Segar

NO	ASM	Contrast	Energy	Homogeneity	Correlation	Kelas Target	Kelas Prediksi
28	-1.432	1.56	-1.329	-1.312	-0.185	Segar	Segar
29	1.071	-0.495	0.913	0.825	-0.077	Segar	Segar

4.7 Evaluasi model

Berdasarkan hasil proses klasifikasi yang telah dilakukan terhadap data testing yang ada, maka dapat diperoleh confusion matrix yang bertujuan untuk melakukan evaluasi dan analisis kinerja dari algoritma atau pendekatan yang digunakan dalam penelitian ini, adapun confusion matrix yang dihasilkan dari proses klasifikasi yang telah dilakukan adalah sebagai berikut :

Tabel 4.9 Hasil Confussion Matrix

Kelas Target	Kelas Prediksi	
	Segar	Tidak Segar
Segar	18	0
Tidak Segar	2	9

Untuk melakukan evaluasi kinerja dari model atau algoritma yang digunakan dalam penelitian ini, diperlukan beberapa informasi nilai kinerja dari hasil proses klasifikasi yang telah dilakukan diantaranya :

4.7.1 Recall

Fungsi recall dalam confusion matrix dapat dihitung dengan persamaan berikut :

$$Recall = \frac{TP}{TP + FN}$$

Sehingga, dengan persamaan tersebut, maka nilai recall berdasarkan confusion matrix diatas adalah sebagai berikut :

$$Recall = 18 / (18 + 0) = 1.0$$

4.7.2 Precision

Fungsi precision pada confusion matrix dapat di hitung dengan persamaan dibawah ini,

$$Precision = \frac{TP}{TP + FP}$$

Sehingganya dengan persamaan tersebut maka nilai precision berdasarkan confusion matrix diatas adalah sebagai berikut :

$$\begin{aligned} \text{Precision} &= 18 / (18 + 2) \\ &= 0.9 \end{aligned}$$

4.7.3 Accuracy

Fungsi Accuracy pada confusion matrix dapat dihitung dengan persamaan dibawah ini,

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

Sehingganya berdasarkan persamaan diatas, maka nilai akurasi diperoleh sebagai berikut :

$$\begin{aligned} \text{Akurasi} &= (18 + 9) / (18 + 9 + 0 + 2) \\ &= 0.93 \end{aligned}$$

4.7.4 F1 Score

Fungsi F1 Score dapat dihitung dengan menggunakan persamaan berikut :

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Berdasarkan persamaan diatas, maka nilai F1 Score diperoleh sebagai berikut :

$$\begin{aligned} \text{F1 Score} &= 2 * ((0.9 * 1.0) / (0.9 + 1.0)) \\ &= 2 * (0.9 / 1.9) \\ &= 2 * 0.47 \\ &= 0.95 \end{aligned}$$

BAB V

PEMBAHASAN HASIL

5.1 Objek

Udang yang dikumpulkan diamati secara mandiri dan diidentifikasi menggunakan Metode Gray Level Co-occurrence Matrix dan ANN. Udang yang diamati dari udang segar sampai sudah tidak segar. Udang yang segar kondisi kulit yang masih bening, warna daging pada udang sedikit keabu-abuan dan Udang tidak segar cenderung berwarna merah dari bagian kepala hingga ekornya.

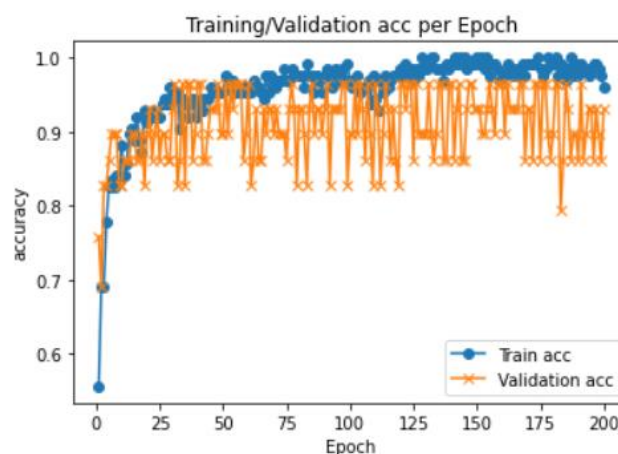
5.2 Pembahasan hasil eksperimen

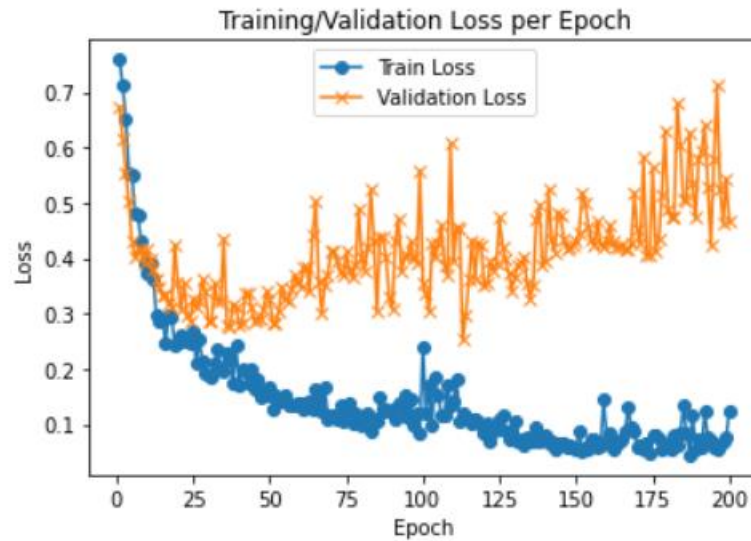
Berdasarkan perolehan hasil percobaan yang telah dilakukan pada bab sebelumnya, maka dapat dijelaskan beberapa hal sebagai berikut :

- a. Data gambar yang berhasil dikumpulkan pada penelitian ini adalah sebanyak 142 gambar udang, yang dibagi kedalam dua buah kelas yakni kategori udang segar dan tidak segar. Berdasarkan visualisasi citra bahwa kedua jenis kategori udang tersebut dapat dibedakan pada sudut pandang tekstur warna pada permukaan udang.
- b. Pada percobaan yang dilakukan di penelitian ini fitur citra di ekstraksi menggunakan metode GLCM dengan menghasilkan 5 fitur statistik dari citra, sehingga dapat mengurangi waktu komputasi dari algoritma klasifikasi ANN yang digunakan pada penelitian ini sehingga penggunaan algoritma ANN ini menjadi lebih efisien untuk diterapkan pada aplikasi yang akan dikembangkan kedepannya.
- c. Dalam penggunaan metode GLCM untuk ekstraksi fitur digunakan parameter nilai angle 1^0 dan parameter nilai distance adalah 1 pixel, kemudian metode ini memanfaatkan 5 nilai fitur statistik sebagai hasil ekstraksi fitur diantaranya nilai Contrast, Energy, Homogeneity, Angular

Second Moment (ASM) dan Correlation. Kelima fitur statistik tersebut akan bertindak sebagai nilai yang mewakili setiap citra untuk kemudian menjadi variabel input pada algoritma klasifikasi yang digunakan.

- d. Pada percobaan yang dilakukan, algoritma ANN yang digunakan sebagai pengklasifikasi data citra digunakan arsitektur sebagaimana yang telah digambarkan pada bab sebelumnya yang terdiri atas 5 nilai input, kemudian terdiri dari 3 hidden layer, untuk hidden layer 1 terdiri dari 512 neuron, hidden layer 2 terdiri dari 1024 neuron dan hidden layer 3 terdiri atas 1024 neuron, untuk setiap neuron pada hidden layer digunakan fungsi aktivasi Relu, sedangkan pada output layer digunakan fungsi aktivasi sigmoid dan digunakan fungsi optimasi Adam pada arsitektur tersebut.
- e. Dalam proses training data menggunakan algoritma ANN dilakukan proses tuning parameter guna menghasilkan parameter yang tepat dari ANN untuk mendapatkan model terbaik berdasarkan data yang digunakan. Pada proses tuning parameter dihasilkan bahwa model dapat berkinerja baik pada nilai learning rate sebesar 0.001, jumlah epochs sebanyak 200.
- f. Berdasarkan hasil proses learning data yang dilakukan, diperoleh nilai akurasi untuk data training yang berkisar di angka 96% dan nilai akurasi untuk data validasi berkisar di angka 93%, sementara untuk nilai loss untuk data training berada di angka 1% dan nilai loss untuk data validasi berada di angka 35%. Sebagaimana yang ditunjukkan pada grafik di bawah ini.





Grafik nilai loss training dan validasi

- g. Berdasarkan hasil proses klasifikasi yang dilakukan terhadap data testing yang digunakan yang berjumlah sebanyak 29 data gambar, dapat diperoleh nilai kinerja model dari algoritma ANN yang digunakan yakni nilai recall sebesar 100%, nilai precission sebesar 90%, nilai accuracy sebesar 93% dan nilai F1 Score sebesar 95%. Sehingga berdasarkan perolehan nilai kinerja model tersebut menunjukkan bahwa penggunaan algoritma ANN dapat bekerja dengan baik ketika digunakan bersamaan dengan penggunaan metode GLCM sebagai metode ekstraksi fitur, sehingga penggunaan kedua metode tersebut secara bersamaan dapat digunakan dalam mengembangkan sebuah aplikasi pengidentifikasi kualitas suatu udang untuk dapat dimanfaatkan di dunia industri.

Tabel 5.1 Nilai kinerja metode yang digunakan

Recall	Precision	Accuracy	F1 Score
100%	90%	93%	95%

BAB VI

PENUTUP

6.1 Kesimpulan

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

- a. Dapat diketahui kinerja penggunaan metode GLCM dan ANN dalam melakukan proses identifikasi kualitas udang, bahwa penggunaan metode tersebut memiliki kinerja sangat baik ditunjukkan dengan perolehan nilai akurasi sebesar 93%, recall sebesar 100%, precision sebesar 90% dan nilai F1 Score sebesar 95%. Berdasarkan hasil training sehingganya metode yang digunakan dapat direkomendasikan untuk mengembangkan sebuah sistem identifikasi kualitas udang agar dapat diterapkan di dunia industri.
- b. Berdasarkan hasil percobaan dan pembahasan yang telah uraikan sebelumnya dapat diketahui hasil proses identifikasi kualitas udang mampu memberikan output yang sangat baik, dengan merujuk hasil pengukuran kinerja dari metode yang digunakan.

6.2 Saran

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

- a. Perlu dilakukan pengumpulan data yang lebih baik lagi terutama jumlah data yang digunakan pada penelitian ini sangat terbatas, sehingganya ada peluang untuk meningkatkan kinerja model menjadi lebih baik agar dapat terimplementasi dengan baik ketika dilakukan pengembangan sistem untuk digunakan di dunia industry.
- b. Perlu dilakukan riset lebih lanjut untuk meningkatkan kinerja algoritma yang digunakan karena pada pembahasan hasil masih terlihat nilai loss

pada penggunaan data uji masih jauh lebih besar dari pada nilai loss pada data training.

- c. Diharapkan kedepannya dapat digunakan pendekatan deep learning untuk dapat meningkatkan kinerja model yang diperoleh tentunya setelah dilakukan proses penyempurnaan dataset yang digunakan pada penelitian terutama dari segi banyaknya jumlah data yang digunakan.

DAFTAR PUSTAKA

- Herviana Masrani, Ilhamsyah, & Ikhwan Ruslianto. (2018). APLIKASI PENGENALAN POLA PADA HURUF TULISAN TANGAN MENGGUNAKAN JARINGAN SARAF TIRUAN DENGAN METODE EKSTRAKSI FITUR GEOMETRI. *Jurnal Coding, Sistem Komputer Untan*, 69-78.
- R. Hadapiningradja Kusumodestoni, & Akhmad Khanif Zyen. (2015). PREDIKSI KECEPATAN ANGIN MENGGUNAKAN MODEL NEURAL NETWORK UNTUK MENGHETAHUI BESAR DAYA LISTRIK YANG DIHASILKAN. *Jurnal DISPROTEK*, 53 - 59.
- R. Hadapiningradja Kusumodestoni, & Akhmad Khanif Zyen. (2015). PREDIKSI KECEPATAN ANGIN MENGGUNAKAN MODEL NEURAL NETWORK UNTUK MENGHETAHUI BESAR DAYA LISTRIK YANG DIHASILKAN. *Jurnal DISPROTEK*, 53 - 59.
- Raynaldi Fatih Amanullah, Ade Pujianto, Bayu Trisna Pratama, & Kusrini. (2018). DETEKSI MOTIF BATIK MENGGUNAKAN EKSTRAKSI TEKSTUR DAN JARINGAN SYARAF TIRUAN. *Seminar Nasional Teknologi Informasi dan Multimedia*, 10-31 - 10-36.
- Rosa Andrie Asmara, Dwi Puspitasari, Siti Romlah, Qonitatul H, & Robertus Romario. (2017). IDENTIFIKASI KESEGERAN DAGING SAPI BERDASARKAN CITRANYA DENGAN EKSTRAKSI FITUR WARNA DAN TEKSTURNYA MENGGUNAKAN METODE GRAY LEVEL COOCCURRENCE MATRIX. *Prosiding SENTIA*, 89 - 94.
- Rudiati Evi Masithoh, Budi Rahardjo, Lilik Sutiarto, & Agus Hardjoko. (2013). PENGEMBANGAN COMPUTER VISION SYSTEM SEDERHANA UNTUK MENENTUKAN KUALITAS TOMAT. *AGRITECH*, 116-123.
- Syafitri, N. (2011). Pengenalan Pola Untuk Deteksi Uang Koin. *SNTIKI*, 18-24.

Yaltha Rullist, Budhi Irawan, & Andrew Brian Osmond. (2015). APLIKASI IDENTIFIKASI MOTIF BATIK MENGGUNAKAN METODE EKSTRAKSI. *e-Proceeding of Engineering*, 3684 - 3692.

Yaltha Rullist, Budhi Irawan, & Andrew Brian Osmond. (2015). APLIKASI IDENTIFIKASI MOTIF BATIK MENGGUNAKAN METODE EKSTRAKSI FITUR GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM) BERBASIS ANDROID. *e-Proceeding of Engineering*, 3684 - 3692.

KODE PROGRAM

```
from skimage import feature

import os

import cv2

from imutils import paths

import numpy as np

import pandas as pd

from skimage.filters import threshold_otsu

from matplotlib import pyplot as plt

from sklearn.metrics import accuracy_score, ConfusionMatrixDisplay,
confusion_matrix


folder_train = "gambar_udang"


def get_path(folder_train):

    array_path = []

    for path in paths.list_images(folder_train):

        array_path.append(path)

    return array_path


def pra_pengolahan(img_path):

    img = cv2.imread(img_path)
```

```

img = cv2.resize(img,(256,256))

rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

gray = cv2.cvtColor(rgb, cv2.COLOR_RGB2GRAY)

kernel = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]])

result = cv2.filter2D(gray, -1, kernel)

#plt.imshow(img)

#plt.title("hasil pre")

#plt.show()

return result, img

```

```

def segmentasi(impath):

```

```

    im_preprocessing, img = pra_pengolahan(impath)

    _,thresh = cv2.threshold(im_preprocessing, np.mean(im_preprocessing), 255,
                             cv2.THRESH_BINARY_INV)

    edges = cv2.dilate(cv2.Canny(thresh,0,255),None)

    cnt = sorted(cv2.findContours(edges, cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE)[-2],
               key=cv2.contourArea)[-1]

    mask = np.zeros((256,256), np.uint8)

    masked = cv2.drawContours(mask, [cnt],-1, 255, -1)

    dst = cv2.bitwise_and(img, img, mask=mask)

    segmented = cv2.cvtColor(dst, cv2.COLOR_BGR2RGB)

    return segmented

```



```

def ekstraksi_ciri(img, radius, angle):

    img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

    glcm = feature.greycomatrix(img, [radius], [angle], symmetric=False,
normed=True)

    asm = feature.greycoprops(glcm, 'ASM')[0,0]

    con = feature.greycoprops(glcm, 'contrast')[0,0]

    ene = feature.greycoprops(glcm, 'energy')[0,0]

    hmg = feature.greycoprops(glcm, 'homogeneity')[0,0]

    cor = feature.greycoprops(glcm, 'correlation')[0,0]

    return [asm, con, ene, hmg, cor]

```

```

def ekstraksi_data_training(folder_train, radius, angle):

    feature_training = []

    impaths = get_path(folder_train)

    for item in impaths:

        #hasil_prapengolahan = pra_pengolahan(item)

        hasil_segmentasi = segmentasi(item)

        hasil_ekstraksi = ekstraksi_ciri(hasil_segmentasi, radius, angle)

        feature_training.append(hasil_ekstraksi)

    return feature_training

```

```

def ekstraksi_data_testing(self, imgtest, radius, angle):

```

```
feature_testing = []

hasil_prapengolahan = self.pra_pengolahan(imgtest)

hasil_segmentasi = self.segmentasi(hasil_prapengolahan)

hasil_ekstraksi = self.ekstraksi_ciri(hasil_segmentasi, radius, angle)

feature_testing.append(hasil_ekstraksi)

return feature_testing
```

```
def getLabel(folder_train):

    label = []

    impaths = get_path(folder_train)

    for item in impaths:

        label.append(os.path.split(os.path.dirname(item))[-1])

    return label
```

```
label = getLabel(folder_train)
```

```
data = ekstraksi_data_training(folder_train, 1, 0)
```

```
#from tensorflow.keras.utils import to_categorical
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import LabelEncoder, StandardScaler
```

```
label_encoder = LabelEncoder()
```

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

```
normalisasi = StandardScaler()
```

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

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

```
import pandas as pd
```

```
import numpy as np
```

```
pd.DataFrame(np.asarray(x_train)).to_csv('data.csv')
```

```
from imblearn.over_sampling import SMOTE
```

```
oversample = SMOTE()
```

```
X, Y =oversample.fit_resample(x_train, y_train)
```

```
from tensorflow.keras.models import Sequential
```

```
from tensorflow.keras.layers import Dense, Dropout
```

```
from tensorflow.keras.optimizers import SGD, Adam
```

```
from tensorflow.keras import regularizers
```

```
model = Sequential()
```

```

model.add(Dense(512, activation='relu', kernel_regularizer=regularizers.l2(0.01),
               input_shape=(5,)))
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer=Adam(learning_rate=0.001),
              loss='binary_crossentropy',
              metrics=['accuracy'])

history = model.fit(X, Y, validation_data=(x_val, y_val), epochs=200, verbose=1)

import matplotlib.pyplot as plt

f, ax = plt.subplots()

ax.plot([None] + history.history['accuracy'], 'o-')
ax.plot([None] + history.history['val_accuracy'], 'x-')

ax.legend(['Train acc', 'Validation acc'], loc = 0)
ax.set_title("Training/Validation acc per Epoch")
ax.set_xlabel('Epoch')

```

```
ax.set_ylabel('accuracy')
```

```
import matplotlib.pyplot as plt
```

```
f, ax = plt.subplots()
```

```
ax.plot([None] + history.history['loss'], 'o-')
```

```
ax.plot([None] + history.history['val_loss'], 'x-')
```

```
ax.legend(['Train Loss', 'Validation Loss'], loc = 0)
```

```
ax.set_title('Training/Validation Loss per Epoch')
```

```
ax.set_xlabel('Epoch')
```

```
ax.set_ylabel('Loss')
```

```
model_json = model.to_json()
```

```
with open("model.json", "w") as json_file:
```

```
    json_file.write(model_json)
```

```
model.save_weights("model.h5")
```

```
import pickle
```

```
with open('labelencode_model.pkl', 'wb') as file_label:
```

```
    pickle.dump(label_encoder, file_label)
```

```
with open('feature_scaling_model.pkl', 'wb') as file_scaling:
```

```
    pickle.dump(normalisasi, file_scaling)
```

```
with open('feature_validation.pkl', 'wb') as file_fv:
```

```
    pickle.dump(x_val, file_fv)
```

```
with open('label_validation.pkl', 'wb') as file_lv:
```

```
    pickle.dump(y_val, file_lv)
```

```
from sklearn.metrics import accuracy_score, ConfusionMatrixDisplay,  
confusion_matrix
```

```
y_pred = model.predict(x_val)
```

```
y_target = label_encoder.inverse_transform(y_val)
```

```
y_aktual = label_encoder.inverse_transform(np.where(y_pred > 0.5, 1,0).ravel())
```

```
cm = confusion_matrix(y_target, y_aktual, labels=['Segar', 'Tidak Segar'])
```

```
disp = ConfusionMatrixDisplay(cm, display_labels=['Segar', 'Tidak Segar'])
```

```
disp.plot()
```

```
from sklearn.metrics import classification_report
```

```
print(classification_report(y_target, y_aktual))
```

```
import pandas as pd
```

```
import numpy as np
```

```
pd.DataFrame(np.asarray(y_aktual)).to_csv('hasil_klasifikasi.csv')
```

```
pd.DataFrame(np.asarray(y_target)).to_csv('kelas_target.csv')
```

```
pd.DataFrame(np.asarray(x_val)).to_csv('feature_data_test.csv')
```

Daftar Riwayat Hidup



Nama : Wanda Aprilia Pangemanan
Nim : T3115115
Jenis Kelamin : Perempuan
Tempat Tanggal Lahir : Gorontalo, 15 April 1998
Status : Menikah
Alamat : Jl. Bilinggata Kel. Dulomo Selatan
Kewarganegaraan : Indonesia
Agama : Islam
Email : pangemananw2@gmail.com

Riwayat Pendidikan :

1. Tamatan SD Negeri 47 hulonthalangi pada tahun 2009
2. Tamatan SMP Negeri 6 Gorontalo 2012
3. Tamatan SMA Negeri 1 Gorontalo 2015
4. Masuk dan diterima di Universitas Ichsan Gorontalo Pada tahun 2015

