

**IDENTIFIKASI TINGKAT KEMATANGAN BUAH
NAGA MENGGUNAKAN METODE ANN
(*Artificial Neural Network*)**

Oleh

SITI NURKHOLIZA ANWAR

T3116022

SKRIPSI

Untuk Memenuhi Salah Satu Syarat Ujian

Guna Memperoleh Gelar Sarjana



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

**IDENTIFIKASI TINGKAT KEMATANGAN BUAH
NAGA MENGGUNAKAN METODE ANN
(*Artificial Neural Network*)**

**Oleh
SITI NURKHOLIZA ANWAR
T3116022**

SKRIPSI

**Untuk Memenuhi Salah Satu Syarat Ujian
Guna Memperoleh Gelar Sarjana**



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

PERSETUJUAN UJUAN SKRIPSI

**IDENTIFIKASI TINGKAT KEMATANGAN BUAH
NAGA MENGGUNAKAN METODE ANN
(*Artificial Neural Network*)**

OLEH

Siti Nurkholiza Anwar

T3116022

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

Gorontalo, Desember 2020

Pembimbing I



Asmaul Husna, M.Kom
NIDN.0911108602

Pembimbing II



Yulianty Lasena, M.Kom
NIDN.0907078603

PENGESAHAN SKRIPSI
IDENTIFIKASI TINGKAT KEMATANGAN BUAH
NAGA MENGGUNAKAN METODE ANN
(Artificial Neural Network)

Oleh

Siti Nurkholiza Anwar

T3116022

Di Periksa oleh Panitia Ujian S1

Universitas Ichsan Gorontalo

1. Ketua Penguji
Amiruddin, M.Kom



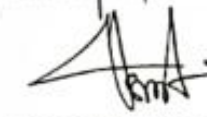
2. Anggota Penguji
Sudirman S. Panna, M.Kom



3. Anggota Penguji
Abd. Rahmat Karim Haba, M.Kom



4. Anggota Penguji
Asmaul Husna, M.Kom



5. Anggota Penguji
Yulianty Lasena, M.Kom



PERNYATAAN SKRIPSI

Dengan ini saya menyatakan bahwa :

1. Karyatulis (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. Karyatulis (Skripsi) saya ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali dari arahan Tim Pembimbing.
3. Dalam karyatulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah dipublikasi orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pula dalam daftar pustaka.
4. Pernyataan sisanya 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, Desember 2020

Yang Membuat Pernyataan,



Siti Nurkholiza Anwar

ABSTRAK

Identifikasi tingkat kematangan buah naga dalam penelitian ini dibagi menjadi dua kelompok kematangan buah naga yaitu kelompok mentah dan kelompok matang. Penelitian ini bertujuan untuk dapat mengklasifikasi tingkat kematangan berdasarkan citra buah naga dengan menggunakan metode fitur ekstraksi yaitu *gray level co-occurrence matrix* (GLCM). Metode penelitian ini terdiri dari: konversi data rgb ke grayscale, normalisasi citra, deteksi kematangan buah naga, ekstraksi fitur dan identifikasi. Pengumpulan data yang diambil dari data real berjumlah 60 image yang dipakai dalam penelitian ini terdiri 40 data training dan 20 data testing yang merupakan data image RGB dengan format JPG. Masing-masing data terdiri dari 2 kategori kematangan, untuk data training terdiri dari 20 image buah naga matang 99% dan 20 image buah naga matang 85%. Sedangkan untuk data testing terdiri dari 10 image buah naga matang 99% dan 10 image buah naga matang 85%. Data citra tersebut diolah menjadi citra grayscale yang kemudian dilakukan deteksi kematangan buah naga. Setelah didapat kematangan buah naga kemudian dilakukan segmentasi pada bagian lokasi buah naga yang ditemukan. Selanjutnya dilakukan perhitungan ciri menggunakan *Gray Level Co-Occurrence Matrix* (GLCM). Algoritma yang digunakan untuk proses identifikasi adalah algoritma *Artificial Neural Network* (ANN). Hasil akhir pengujian menunjukkan bahwa metode yang diusulkan telah mampu mendeteksi kematangan buah naga dengan hasil akurasi = $9/10 * 100\% = 90\%$ yang dihitung menggunakan confusion matrix Dengan demikian penerapan metode *Gray Level Co-Occurrence Matrix* dan *Artificial Neural Network* pada masalah tingkat kematangan buah naga perlu dikembangkan.

Kata kunci : Buah Naga, *Gray Level Co-Occurrence Matrix*, *Artificial Neural Network*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah, segala puja dan puji syukur penulis panjatkan atas Rahmat Allah SWT sehingga penulis dapat menyelesaikan skripsi ini yang berjudul **“IDENTIFIKASI TINGKAT KEMATANGAN BUAH NAGA MENGGUNAKAN METODE ANN (*Artificial Neural Network*)”** untuk memenuhi salah satu syarat Ujian Akhir guna memperoleh gelar Sarjana Komputer pada 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 terimakasih dan penghargaan yang setinggi-tingginya kepada:

1. Bapak Mohammad Ichsan Gaffar, S.E, M.Ak, selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo;
2. Bapak Dr. Abdul Gaffar La Tjokke, M.Si, selaku Rektor Universitas Ichsan Gorontalo;
3. Ibu Zohrahayati, S.Kom., M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
4. Bapak Sudirman Panna, M.Kom, selaku Wakil Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
5. Ibu Irma Surya Kumala Idris, M.Kom, selaku Wakil Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
6. Bapak Sudirman Melangi, M.Kom, selaku Wakil Dekan III Bidang Kemahasiswaan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
7. Bapak Irvan Abraham Salihi, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;

8. Ibu Asmaul Husna M.Kom, selaku Pembimbing I yang telah banyak membantu penyusunan skripsi ini;
9. Ibu Yulianty Lasena M.Kom, selaku Pembimbing II yang telah banyak membantu penyusunan skripsi ini;
10. Bapak dan ibu dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis;
11. Kedua orang tua saya yang tercinta (Anwar Bimbing dan Misna Salote S.Pd), atas segala kasih sayang, jerih payah dan doa restunya dalam membesarkan dan mendidik penulis;
12. Kepada Kakak dan adik-adik saya yang tercinta (Alm.Khoirul Aksal Bimbing, Siti Nuriasi Anwar, Siti Khoirun Nisa Bimbing) serta keluarga besar (Bimbing, Salote) yang telah memberikan dukungan kepada penulis;
13. Kepada kerabat dekat dan saudara-saudaraku (Yusran Moha, Merlian, Englita, Santri, Wita, Fita, Gina, Yayan Hadia, Randi Salote, Yayu Gobel, Gita Gobel, Alma, Aldin Laudengi) nenek tercinta (Hasbia Nakoda), sahabat-sahabatku (Sela, Ica, Gita, Dian, Indi, Elin, Fian, Sukma, Windri, Kina, Hasrin, Egi, Aldi, Aldo, Ifan, Awin, Wanda, Naldi) dan masih banyak lagi yang tidak bisa saya sebutkan satu persatu, Keluarga Besar P2MA (Persatuan Pelajar Mahasiswa Atinggola) serta Orang terdekatku yang tersayang.
14. Teman-teman KKLP Desa Talumopatu Kecamatan Mootilango Kabupaten Gorontalo dan Kepala Desa beserta masyarakat desa Talumopatu yang tidak dapat saya sebutkan satu persatu ucapan terima kasih atas semua support.
15. Rekan-rekan seperjuangan teristimewa (Nuraini, Yolanda, Sarintan, Serli, Sesi, Wahyu, Dana, Alfin, Karel) yang telah banyak memberikan bantuan dan dukungan moril yang sangat besar kepada penulis;
16. 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 mereka kepada kami. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat kekurangan. Oleh karena itu, penulis

sangat mengharapkan adanya kritik dan saran yang konstruktif. Akhirnya penulis berharap semoga hasil yang telah dicapai ini dapat bermanfaat bagi kita semua, Aamiin.

Gorontalo, Desember 2020

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
PERSETUJUAN UJUIAN SKRIPSI	iii
PENGESAHAN SKRIPSI.....	iv
PERNYATAAN SKRIPSI.....	v
<i>ABSTRACT</i>	vi
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL	xv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Identifikasi Masalah.....	3
1.3. Rumusan Masalah.....	4
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	4
BAB II LANDASAN TEORI	5
2. 1. Tinjauan Studi	5
2.2 Tinjauan Pustaka	7
2.2.1 Buah Naga.....	7
2.2.2 Pengolahan Citra.....	8
2.2.3 GLCM (<i>Gray Level Co-occurrence Matrix</i>)	10
2.2.4 Proses <i>Grayscale</i>	11
2.2.5 ANN (<i>Artificial Neural Network</i>)	11
2. 3 Kerangka Pikirs.....	18
BAB III METODE PENELITIAN	19
3.1 Jenis, Metode, Subjel, Objek, dan Lokasi Penelitian.....	19
3.2 Pengumpulan Data	20
3.3 Pemodelan	20

3.3.1.	Pra Pengolahan	21
3.3.2.	Ekstrasi Fitur Warna.....	21
3.3.3.	Data <i>Training</i>	21
3.3.4.	<i>Training</i> Menggunakan ANN.....	21
3.3.5.	Model.....	21
3.3.6.	Data Testing.....	22
3.3.7.	Hasil Klasifikasi.....	22
3.4	Evaluasi	22
BAB IV HASIL PENELITIAN.....		23
4.1	Hasil Pengumpulan Data	23
4.2	Hasil Pemodelan	23
4.2.1.	PraPengolahan	23
4.2.2.	Citra <i>segmentasi</i>	25
4.2.3.	Metode <i>Otsu Thresholding</i>	25
4.2.4.	Ekstraksi Fitur	26
4.2.5.	Klasifikasi	30
4.3	Evaluasi	39
4.3.6.	<i>Confusion Matrix</i>	39
BAB V PEMBAHASAN		41
5.1	Pembahasan Pengumpulan Data	41
5.2	Pembahasan Model	41
5.3	Pembahasan System.....	42
BAB VI PENUTUP		44
6.1	Kesimpulan	44
6.2	Saran.....	44
DAFTAR PUSTAKA.....		45

DAFTAR GAMBAR

Gambar 2.1: Buah Naga [1]	8
Gambar 2.2: Tingkat Keabuan Pada Grayscale [1]	11
Gambar 2.3: Model Neuron [12]	12
Gambar 2.4: Struktur ANN [12]	12
Gambar 2.5: Inisialisasi Bobot [13]	14
Gambar 3.1: Desain Penelitian	19
Gambar 3.1: Pengumpulan identifikasi	20
Gambar 4.1 Citra Segmentasi.....	26
Gambar 4.2 <i>Arsitektur Jaringan Backpropagation</i>	30
Gambar 5.1 :Tampilan Sistem	43
Gambar 5.2 Input gambar training.....	43
Gambar 5.3 Input gambar testing	43
Gambar 5.4 Input nilai	44
Gambar 5.5 Prapengolahan	44
Gambar 5.6 Hasil	44

DAFTAR TABEL

Tabel 2.1: State of the art	5
Tabel 2.2: Contoh Dataset	13
Tabel 4.1 Pengumpulan Data Set.....	23
Tabel 4.2 Hasil output dari RGB, <i>Grayscale</i> , <i>Histogram Equalization</i>	25
Tabel 4.3 Nilai dari Suatu Citra	26
Tabel 4.4 Matrix GLCM	27
Tabel 4.5 <i>Matrix Ternormalisasi</i>	27
Ttabel 4.6 Hasil Nilai Fitur Ekstraksi	30
Tabel 4.7 Hasil Data Training.....	31
Tabel 4.8 Hasil Data Testing	32
Tabel 4.9 Hasil Confusion Matrix	40

BAB I

PENDAHULUAN

1.1. Latar Belakang

Menurut Renasari. Novita. (2010) Buah Naga memiliki beberapa spesies buah yang berbeda, klasifikasi buah naga termasuk kelompok tanaman kaktus atau *family Cactaceae* dan *subfamily Hylocereae*. Varietas buah naga sejauh ini dibedakan kedalam empat grup berdasarkan warna buah, yaitu buah naga daging putih (*Hylocereus undatus*), buah naga daging merah (*Hylocereus polyrhizus*), buah naga daging super merah (*Hylocereus costaricensis*), dan buah naga kulit kuning daging putih (*Selenicereus megalanthus*). Tanaman buah naga mempunyai batang sulur yang tumbuh menjalar, berwarna hijau dengan bentuk segi tiga. Bunganya berukuran besar, berwarna putih-kuning muda, dan mekar umumnya di malam hari. Setelah bunga layu maka terbentuk bakal buah yang menggelayut disetiap batang.[1]

Saat ini masyarakat juga sudah tidak asing lagi dengan *hylocereus costaricensis* (buah naga merah), yang dapat ditemukan dipasar maupun suwalayan. Namun masalah yang terjadi adalah kita terkadang mendapati buah naga yang dagingnya terasa hambar diakibatkan proses panen yang belum tepat waktu atau dipanen sebelum matang dan hanya diamati secara visual manusia yang terkadang tidak konsisten dikarenakan faktor kelelahan dan perbedaan pendapat pada masing-masing pengamat, maka dari itu bidang *computer vision* sendiri terus berusaha mencari cara agar komputer dapat menerima cara kerja indra manusia.

Pada penelitian sebelumnya telah dilakukan identifikasi kematangan buah naga menggunakan metode DCO (*Discrete Cosine Transform*). Dan prosentase akurasi yang didapatkan yaitu 80% dengan perbandingan data *training* dan data *testing* sebanyak 60 : 10, prosentase akurasi 60% dengan perbandingan data *training* dan data *testing* sebanyak 45 : 15, prosentase akurasi 15% dengan perbandingan data *training* dan data *testing* sebanyak 30 : 20, dan prosentase akurasi 45% dengan perbandingan data *training* dan data *testing* sebanyak 15 : 25. Sehingga besarnya tingkat akurasi dipengaruhi oleh data *training* yang digunakan.[1]

Menurut Zheng dkk (2010) GLCM (*Gray Level Co-occurrence Matrix*) merupakan metode tekstur statistik yang banyak digunakan dimana ciri tekstur diekstraksi dengan beberapa pendekatan statistik dari matriks *co-occurrence*. Siqueira dkk (2013) menyatakan dalam analisis citra, GLCM telah terbukti menjadi deskriptor tekstur yang paling kuat untuk klasifikasi data dan masla regresi.[2]

Jahns dkk (2001) berpendapat bahwa dalam analisis citra untuk produk makanan, warna juga merupakan atribut yang berpengaruh memberikan informasi visual dan deskripsi yang kuat untuk melakukan pengukuran. Ciri warna dari sebuah objek dapat diekstraksi dengan memeriksa setiap piksel dalam baris objek, dan telah terbukti sukses untuk melakukan pengukuran objektif dari berbagai jenis makanan mulai dari buah, biji-bijian, daging dan sayuran[2]. Maka dari itu penelitian ini akan menggunakan metode pengolahan citra yaitu GLCM (*Gray Level Co-occurrence Matrix*) karena metode ini adalah yang paling sering digunakan untuk analisis tekstur yang didasarkan pada ciri statistik citra dimana dalam perhitungan statistiknya menggunakan distribusi derajat keabuan (histogram) dengan mengukur tingkat kontras, granularitas, dan kekasaran suatu daerah dari hubungan ketetanggaan antar piksel didalam citra.

Metode Artificial Neural Network (ANN) disebut juga *Simulated Neural Network (SNN)* dan sering disebut juga sebagai Jaringan Syaraf Tiruan (JST). Hal tersebut dikarenakan oleh metode ini merupakan tiruan susunan syaraf (neuron) manusia. ANN adalah sebuah alat pemodelan data statistik dan non statistik. ANN dapat melakukan pemodelan hubungan yang rumit (kompleks) antara input dan output dengan tujuan untuk menemukan pola pada data. [3]. ANN merupakan sistem komputasi yang arsitektur dan operasinya diilhami dari pengetahuan tentang sel syaraf biologis di dalam otak. ANN dapat digambarkan sebagai model matematis dan komputasi untuk fungsi aproksimasi non-linear, klasifikasi data cluster dan regresi non-parametrik atau sebuah simulasi dari koleksi model jaringan syaraf biologi [4]. Pada penelitian yang dilakukan oleh Wahyu Eko Susanto Dwiza Riana menyatakan bahwa nilai akurasi algoritma ANN lebih baik dari pada algoritma K-NN, dengan tingkat akurasi 80,00% untuk algoritma ANN sedangkan

algoritma K-NN tingkat akurasi 71,00% [5]. ANN memiliki keunggulan seperti kemampuan belajar, kemampuan paralel, kemampuan untuk memodelkan fungsi *non linier* dan sifat *fault tolerance*. [6]

Dari penjelasan-penjelasan tersebut maka peneliti akan melakukan uji coba terhadap metode ANN (*Artificial Neural Network*) dalam mengidentifikasi kematangan buah naga untuk mengukur matang tidaknya buah dari warna kulit buah seperti buah naga yang belum matang ditandai dengan warna kulit hijau dan memiliki tekstur yang keras. Kulit luar buah naga akan berubah warna seiring bertambahnya kematangan, buah yang sudah matang memiliki warna kulit buah berwarna merah dan merata. Adapun berbagai masalah yang terjadi seperti bagaimana membangun aplikasi mengidentifikasi kematangan buah naga dengan menggunakan media komputer, bagaimana bisa mengetahui matang tidaknya buah naga tersebut, bisakah mengidentifikasi tingkat kematangan buah naga dengan menggunakan metode ANN (*Artificial Neural Network*).

Berdasarkan pemaparan di atas, maka sangat perlu dilakukan penelitian tentang **“IDENTIFIKASI TINGKAT KEMATANGAN BUAH NAGA MENGGUNAKAN METODE ANN (*ARTIFICIAL NEURAL NETWORK*)”**. Sehingga menjadi tolak ukur pengujian metode ANN untuk mengetahui kematangan buah naga yang sempurna.

1.2. Identifikasi Masalah

Berdasarkan latar belakang diatas, maka masalah yang harus di selesaikan yaitu menguji coba metode dalam identifikasi kematangan buah naga yang masih dilakukan secara manual melalui pengamatan visual dan belum efisien.

1.3. Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka peneliti mengambil rumusan masalah “Bagaimana hasil uji coba Metode ANN (*Artificial Neural Network*) dalam mengidentifikasi tingkat kematangan buah naga?”

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan yang ingin di capai dalam penelitian ini adalah “Untuk mengetahui hasil uji coba metode ANN (*Artificial Neural Network*) dalam mengidentifikasi tingkat kematangan buah naga”

1.5. Manfaat Penelitian

Hasil penelitian ini diharapkan dapat memberikan manfaat kepada berbagai pihak sebagai berikut:

1) Manfaat Teoritis

Hasil penelitian ini dapat menambah ilmu pengetahuan dan informasi mengenai sistem komputer dalam menguji coba metode ANN (*Artificial Neural Network*) maupun mengidentifikasi kematangan buah naga berbasis *Computer Vision* dan dapat membantu konsumen maupun produsen.

2) Manfaat Praktis

Manfaat penelitian ini sebagai referensi dan memberikan tambahan pengetahuan mengenai penerapan uji coba metode ANN (*Artificial Neural Network*) terhadap sistem yang dapat mengenali buah naga tersebut telah matang berdasarkan gambar.

BAB II LANDASAN TEORI

2. 1. Tinjauan Studi

Tabel 2.1 State of the art

NO	PENELITI	JUDUL	TAHUN	METODE	HASI
1	Resty wulanningru m dan NadhaVera Wihra Lelitavistara	Discrete Cosine Transform untuuk identifikasi citra Hylocereus costaricensis	2015	<i>Discrete Cosine Transform (DCT)</i>	Pada penelitian ini menggunakan objek buah naga merah. Proses akurasi yang didapatkan yaitu 80% dengan masing-masing buah naga
2	Restu Widodo, Agus Wahyu Widodo, dan Arry Supriyanto	Pemanfaatan Ciri <i>Gray</i> <i>Love Co- Occurrence Matrix</i> (GLCM) Citra Buah Jeruk Keprok (<i>Citrus Reticulata Blance</i>) Untuk Klasifikasi Mutu	2018	GLCM (<i>Gray Level Co- Occurrence Matrix</i>)	Pemanfaatan hasil ekstraksi ciri <i>gray level co-occurrence matrix</i> (GLCM) citra jeruk keprok untuk klasifikasi mutu dapat memberikan hasil akurasi terbaik sebesar 82.5% dengan jumlah data lati sebanyak 20 dan 30 data, pada jeruk ketetanggaan piksel (<i>distance</i>) sebesar 2 dan pada arah GLCM 40°. hasil ini menunjukkan bahwa sistem yang telah

					dibuat mampu mengklasifikasi mutu dengan baik.
3	Mochammad Yusuf Habibi dan Edwin Riksakomara	Peramalan Harga Garam Konsumsi Menggunakan Artificial Neural Network Feedforward-Backpropagation.	2017	<i>Artificial Neural Network Feedforward-Backpropagation</i>	Pada penelitian ini, digunakan metode Artificial Neural Network. Model yang dihasilkan pada penelitian ini memiliki nilai MAPE sebesar 4.394%. Hal tersebut menunjukkan model yang dihasilkan memiliki hasil permalan sangat baik.

2.2 Tinjauan Pustaka

2.2.1 Buah Naga

Buah naga termasuk dalam kelompok tanaman kaktus atau *family Cactaceae dan subfamily Hylocereanea*. Dalam subfamily ini terdapat beberapa genus, sedangkan buah naga termaksud dalam *genus Hylocereus*. Genus ini pun terdiri atas sekitar 16 spesies. Dua di antaranya memiliki buah yang komersial, yaitu *Hylocereus undatus* (berdaging putih) dan *Hylocereus costaricensis* (daging merah). Daerah asal kaktus hutan yang buah berwarna merah dan bersisik ini adalah Meksiko, Amerika Tengah, dan Amerika Utara. Di daerah asalnya tersebut buah naga atau *dragon fruit* ini dinamai *pitahaya* atau *pyata roja*. Tanaman buah naga lebih dikenal sebagai tanaman dari Asia karena dikembangkan secara besar-besaran di Asia seperti di Vietnam dan Thailand. [7]

Buah naga tidak seperti jenis buah lainnya yang hanya panen sekali dalam setahun, buah naga dapat dipanen berkali-kali dalam setahun. Masa berbuah tanaman buah naga biasanya berlangsung selama 4-6 bulan. Dalam masa tersebut, tanaman berbuah dan dipanen bergantian, sehingga buah naga dapat tersedia hampir selama setengah tahun. Masa berbuah tanaman buah naga juga bisa diperpanjang dengan pengaturan pencahayaan. Buah naga relatif lebih awet dibandingkan jenis buah yang lain. Dengan waktu pemanenan yang tepat dan penyimpanan yang baik, buah ini mampu bertahan hingga 1-2 bulan dalam kondisi segar. Faktor-faktor seperti akan kebutuhan buah naga yang semakin meningkat, kandungan gizi, keawetan buah yang tinggi, serta buah yang dapat diolah lebih lanjut mendorong semakin cerah perkembangan buah naga di tanah air.[8]

Hingga kini terdapat empat jenis buah naga diusahakan dan memiliki banyak prospek baik. Salah satunya yakni buah naga merah yang lebih banyak dikenal di China dan Amerika ini memiliki buah dengan kulit berwarna merah dan daging berwarna merah keunguan. Rasa buah naga merah lebih manis dibandingkan dengan buah naga putih, dengan kadar kemanisan mencapai 13-15% Brix. Tanaman ini tergolong jenis tanaman sering berbuah, bahkan cenderung berbunga sepanjang

tahun. Sayangnya tingkat keberhasilan bunga menjadi bunga sangat kecil, hanya mencapai 50%. [8]



Gambar 2.1: Buah Naga [1]

2.2.2 Pengolahan Citra

Pengolahan citra digital adalah pemrosesan citra, khususnya dengan menggunakan komputer sehingga citra kualitasnya menjadi lebih baik dan menghasilkan informasi nilai untuk tipe-tipe 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. [9]

Ada beberapa operasi pengolahan citra yang biasanya dilakukan menurut Jaim, 1989, diantaranya. [10]:

2.2.2.1 Representasi dan pemodelan citra

Bertujuan untuk memberikan gambaran suatu hal. Contohnya adalah bagaimana pencahayaan benda di suatu lokasi dari hasil foto kamera, karakteristik jaringan tubuh dari citra X-Ray, posisi target dari sebuah radar dari citra radar, suhu suatu daerah dari citra inframerah atau medan gravitasi suatu daerah dari citra geofisika.[10]

2.2.2.2 Perbaikan citra

Bertujuan untuk menonjolkan fitur-fitur tertentu yang ada pada citra untuk kepentingan ekstraksi fitur, analisis citra atau menampilkan informasi citratersebut.

Contohnya adalah perbaikan kontras, perbaikan tepi objek, pemberian warna semu, penipisan derau, penajaman, dan sebagainya. [10].

2.2.2.3 Pemugaran citra

Bertujuan untuk menghilangkan atau mengurangi degradasi citra. Mirip dengan perbaikan citra, hanya saja pada pemugaran citra sudah diketahui penyebab degradasi citra. Contohnya adalah menghilangkan kesamaran (*deblurring*) yang disebabkan oleh keterbatasan sensor, usia citra yang sudah tua atau banyaknya derau (*noise*) pada citra. [10]

2.2.2.4 Analisis citra (*image analys*)

Bertujuan untuk menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsi dari citra tersebut. Teknik analisis citra memerlukan fitur-fitur tertentu untuk membantu dalam identifikasi objek. Proses segmentasi terkadang diperlukan untuk memisahkan objek yang diinginkan dari sekelilingnya sehingga selanjutnya dapat dilakukan pengukuran. Contohnya adalah mendeteksi tepi objek, ekstraksi batas (*boundary*), dan representasi daerah (*region*). [10]

2.2.2.5 Rekonstruksi citra

Bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Contohnya adalah gambar 3D yang dihasilkan dari beberapa foto rontgen dengan sinar X. [10]

2.2.2.6 Kompresi citra

Bertujuan untuk mengurangi jumlah bit yang dibutuhkan untuk menyimpan atau mengirim gambar tanpa kehilangan informasi yang diperlukan. Citra berukuran besar akan mempengaruhi penggunaan memori dan lamanya waktu pengiriman. [10]

2.2.3 GLCM (*Gray Level Co-occurrence Matrix*)

Gray Level Co-occurrence Matrix (GLCM) merupakan metode yang digunakan untuk analisis tekstur/ekstraksi ciri. GLCM adalah suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra [11].

Koordinat pasangan piksel memiliki jarak d dan orientasi sudut θ . Jarak direpresentasikan dalam piksel dan sudut direpresentasikan dalam derajat. Orientasi sudut terbentuk berdasarkan empat arah sudut yaitu, 0° , 45° , 90° dan 135° , dan jarak antar piksel sebesar 1 piksel menurut.[11]

Tahapan yang dilakukan pada perhitungan GLCM adalah sebagai berikut:[11]

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

➤ *Contrast* =

$$\sum_i \sum_j (i - j)^2 P_{ij}$$

➤ *Correlation* =

$$\frac{\sum_{i=1}^L \sum_{j=1}^L (i - \mu_i)(j - \mu_j)(GLCM(i, j))}{\sqrt{\sigma_i^2 \sigma_j^2}}$$

➤ *Energy* =

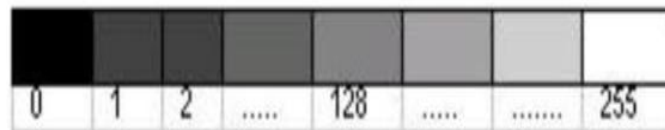
$$\sum_{i,j} P(i, j)^2$$

➤ *Entropy* =

$$\sum_{i,j} P_{ij} \log^2 P_{ij}$$

2.2.4 Proses Grayscale

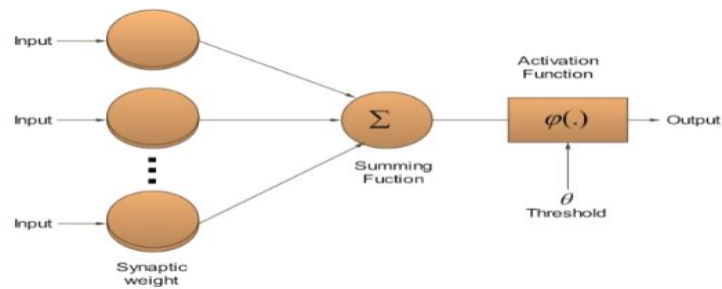
Menurut Putra Darma (2010) dan Nazaruddin Ahmad (2013) adalah *Grayscale* merupakan tahap dari *pre-procesing* yang yang digunakan umntuk mempersiapkan citra agar dapat menghasilkan ciri yang lebih baik pada tahap pemisahan ciri, *Grayscale* juga disebut dengan tingkat keabuan. Pada umumnya warna yang digunakan pada *grayscale* adalah warna hitam sebagai warna minimal (0), dan warna putih (255) sebagai warna maksimal, sehingga warna antaranya adalah abu-abu. Derajat keabuan sendiri memiliki nilai, tidak hanya skala 0 sampai 255. Tergantung pada nilai piksel yang dimiliki oleh citra. [1]



Gambar 2.2: Tingkat Keabuan Pada Grayscale [1]

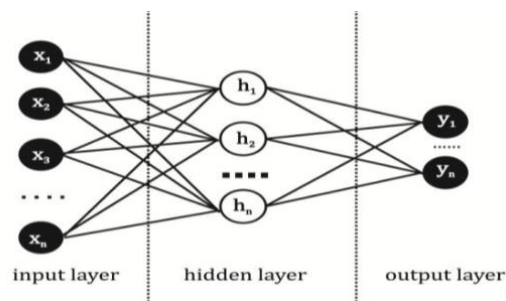
2.2.5 ANN (*Artificial Neural Network*)

Artificial Neural Network (ANN) merupakan model penalaran yang didasarkan pada otak manusia. ANN terdiri dari sejumlah prosesor sangat sederhana dan saling berhubungan yang disebut neuron. Neuron yang terhubung dengan pembobotan (*weight*) melewati sinyal dari neuron satu ke neuron yang lain. ANN mampu mengenalo sinyal input yang agak berbeda dari yang pernah diterima sebelumnya dam mampu bekerja meskipun beberapa neuronya tidak mampu bekerja dengan baik. Jika sebuah neuron rusak maka neuron lain dapat dilihat untuk menggantikan fungsi neuron yang rusak tersebut. Model neuron ANN terdiri; fungsi penjumlah (*summing function*), fungsi aktivasi (*activation function*), dan keluaran (*output*).[11]



Gambar 2.3: Model Neuron [12]

Cara kerja ANN adalah informasi yang masuk (*input*) akan dikirim ke neuron dengan bobot tertentu dan selanjutnya diproses oleh suatu fungsi yang akan menjumlahkan nilai-nilai bobot yang ada. Hasil penjumlahan akan dibandingkan dengan nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Jika input melewati suatu nilai ambang tertentu, maka neuron akan diaktifkan, jika tidak, maka neuron tidak akan diaktifkan, neuron yang diaktifkan akan mengirimkan output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya.[12]



Gambar 2.4: Struktur ANN [12]

Dimana, *inputlayer* ($x_1, x_2, x_3, \dots, x_n$), terdiri dari unit-unit *input*. Unit-unit input menerima pola inputan data dari luar yang menggambarkan suatu permasalahan. *Hidden Layer* ($h_1, h_2, h_3, \dots, h_n$), terdiri dari unit-unit tersembunyi. Dimana outputnya tidak dapat secara langsung diamati. Sedangkan *output layer* ($y_1, \dots, y_n$) terdiri dari unit-unit *output* yang merupakan solusi ANN terhadap suatu permasalahan.[12]

Tahap-tahap pengujian pada model ANN [13], sbb:

1. Tahap 0: Inisiasi bobot sesuai dengan bobot yang telah dihasilkan pada proses pelatihan di atas;
2. Tahap 1: Untuk setiap *input*, lakukan tahap 2-4;
3. Tahap 2: Untuk setiap *input* $i=1, \dots, n$ skalakan bilangan dalam *range* fungsi aktivasi seperti yang dilakukan pada proses pelatihan di atas
4. Tahap 3: Untuk $j=1, \dots, p$:

$$z_{in\ j} = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

$$z_j = f(z_{in\ j})$$

5. Tahap 4: Untuk $k=1, \dots, m$:

$$y_{in\ k} = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

$$y_k = f(y_{in\ k})$$

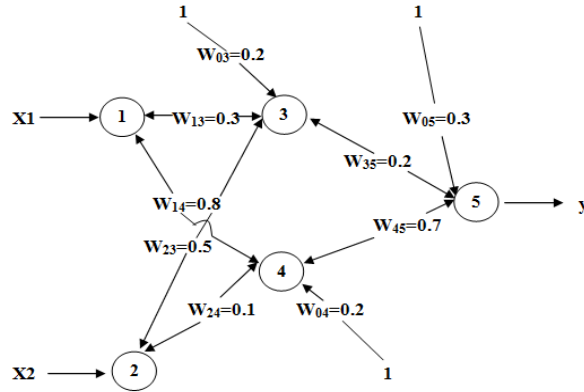
Variable y_k adalah *output* yang masih dalam skala menurut *range* fungsi aktivasi. Untuk mendapatkan nilai *output* yang sesungguhnya, y_k harus dikembalikan seperti semula.

Contoh Penerapan ANN

Untuk memudahkan perhitungan, misalkan jaringan terdiri dari 2 unit input (X1 dan X2) dan satu output yang diambil dari dataset untuk data ke 2, 16, 41 dan 62 [13]

Tabel 2.2: Contoh Dataset [13]

No	X1	X2	Y
2	2	3	0
16	2	6	1
41	1	4	1
62	1	2	0



Gambar 2.5: Inisialisasi Bobot [13]

Kondisi bobot saat inisialisasi seperti ditunjukkan pada gambar di atas dengan jumlah neuron pada layer tersembunyi = 2.

laju pembelajaran (η)= 0.1;

Fungsi aktivasi yang digunakan Sigmoid Biner;

Target *error* = 0.0001 dengan kriteria SSE;

Momentum yang digunakan = 0.95;

Maksimum jumlah iterasi pelatihan = 1,000 kali.

Maka:

1. Nilai pada neuron di hidden layer 3 dan 4:

$$v_j(p) = \sum_{i=1}^n x_i(p)w_{ij}(p)$$

$$y_j(p) = \frac{1}{1 + e^{-v_j(p)}}$$

(p) adalah iterasi ke-i.

$$v_3(1) = x_1w_{13} + x_2w_{23} + 1w_{03} = 2 * 0.3 + 3 * 0.5 + 1 * 0.5 + 1 * 0.2 = 2.3$$

$$y_3(1) = \frac{1}{1+e^{-2.3}} = \frac{1}{1+2.71828^{-2.3}} = 0.908877$$

$$v_4(1) = x_1w_{14} + x_2w_{24} + 1w_{04} = 2 * (-0.8) + 3 * 0.1 + 1 * (-0.2) = 1.7$$

$$y_4(1) = \frac{1}{1 + e^{-1.7}} = 0.845535$$

2. Nilai padaneuron di output layar:

$$v_k(p) = \sum_{j=1}^m x_j(p) w_{jk}(p)$$

$$y_k(p) = \frac{1}{1 + e^{-v_k(p)}}$$

$$v_5(1) = y_3(1)w_{35} + y_4(1)w_{45} + 1w_{05} = 0.908877 * 0.2 + 0.845535 * 0.7 + 1 * 0.3 = 2.3$$

$$y_5(1) = \frac{1}{1 + e^{-2.054411496}} = 0.886392478$$

3. Hitung gradient error untuk neuron pada output layar:

$$e_k(p) = y_{dk}(p) - y_k(p)$$

$$\delta_k(p) = y_k(p) * [1 - y_k(p)] * e_k(p)$$

Untuk data pertama, target/class nilai yang diharapkan adalah $y_d = 0$, sedangkan keluaran yang didapatkan $y_5(1) = 0.886392478$. Maka hitung *error* pada iterasi pertama untuk data pertama pada neuron 5 di output layar.

$$e_1^5(1) = y_d - y_5(1) = 0 - 0.886392478 = -0.886392478$$

$$\delta_5(1) = y_5(1) * (1 - y_5(1)) * e_1^5(1) =$$

$$0.886392478 * (1 - 0.886392478) * (-0.886392478) = -0.08926048$$

4. Hitung koreksi bobot untuk output layar,

$$\Delta w_{jk}(p) = \eta \cdot y_j(p) * \delta_k(p)$$

$$\Delta w_{35} = \eta \cdot y_3(1) * \delta_5(1) = 0.1 * 0.91 * (-0.089) = -0.008112679$$

$$\Delta w_{45} = \eta \cdot y_4(1) * \delta_5(1) = 0.1 * 0.85 * (-0.089) = -0.007547282$$

$$\Delta w_{05} = \eta * 1 * \delta_5(1) = 0.1 * 1 * (-0.089) = -0.0089266048$$

5. Hitung *gradient error* pada hidden layer $\delta_3(1)$ dan $\delta_4(1)$:

$$\delta_j(\mathbf{p}) = y_j(\mathbf{p}) * [(1 - y_j(\mathbf{p}))] + \sum_{k=1}^1 \delta_k(\mathbf{p}) * w_{jk}(\mathbf{p})$$

$$\begin{aligned} \delta_3(1) &= y_3(1) * (1 - y_3(1)) * \sum_{k=1}^1 \delta_k(1) \cdot w_{3k}(1) = y_3(1) * (1 - y_3(1)) * \\ &\delta_5(1) \cdot w_{35}(1) \\ &= 0.91 * (1 - 0.91) = (-0.089) * 0.2 = -0.00147850 \end{aligned}$$

$$\begin{aligned} \delta_4(1) &= y_3(1) * (1 - y_4(1)) * \sum_{k=1}^1 \delta_k(1) \cdot w_{4k}(1) = y_4(1) * (1 - y_4(1)) * \\ &\delta_5(1) \cdot w_{45}(1) \\ &= 0.85 * (1 - 0.85) = (-0.089) * 0.7 = -0.008160558 \end{aligned}$$

6. Hitung koreksi bobot untuk hidden layar

$$\Delta w_{13}, \Delta w_{23}, \Delta w_{03}, \Delta w_{14}, \Delta w_{24}, \text{ dan } \Delta w_{04}$$

$$\Delta w_{ij}(\mathbf{p}) = \eta * x_i(\mathbf{p}) * \delta_j(\mathbf{p})$$

$$\begin{aligned} \Delta w_{13} &= \eta * x_1 * \delta_3(1) = 0.1 * 2 * (-0.0015) = -0.0003 \\ \Delta w_{23} &= \eta * x_2 * \delta_3(1) = 0.1 * 3 * (-0.0015) = -0.00044 \\ \Delta w_{03} &= \eta * 1 * \delta_3(1) = 0.1 * 1 * (-0.0015) = -0.00015 \\ \Delta w_{14} &= \eta * x_1 * \delta_4(1) = 0.1 * 2 * (-0.0082) = -0.00163 \\ \Delta w_{24} &= \eta * x_2 * \delta_4(1) = 0.1 * 3 * (-0.0082) = -0.00245 \\ \Delta w_{04} &= \eta * 1 * \delta_4(1) = 0.1 * 1 * (-0.0082) = -0.00082 \end{aligned}$$

7. Perbaharui bobot untuk neuron pada hidden layar w_{35}, w_{45}, w_{05} :

$$w_{ij}(\mathbf{p} + 1) = w_{ij}(\mathbf{p}) + \Delta w_{ij}(\mathbf{p})$$

$$\begin{aligned} w_{35}(2) &= w_{35}(1) + \Delta w_{35} = 0.2 + (-0.008112679) = 0.191887321 \\ w_{45}(2) &= w_{45}(1) + \Delta w_{45} = 0.7 + (-0.007547282) = 0.692452718 \\ w_{05}(2) &= w_{05}(1) + \Delta w_{05} = 0.3 + (-0.0089266048) = 0.291073952 \end{aligned}$$

8. Perbedaan bobot pada layar tersembunyi, $w_{13}, w_{23}, w_{03}, w_{14}, w_{24}, w_{04}$

$$\begin{aligned} w_{13}(2) &= w_{13}(1) + \Delta w_{13} = 0.3 + (-0.0003) = 0.29974 \\ w_{23}(2) &= w_{23}(1) + \Delta w_{23} = 0.5 + (-0.00044) = 0.499556 \end{aligned}$$

$$\begin{aligned}
w_{03}(2) &= w_{03}(1) + \Delta w_{03} = 0.2 + (-0.00015) = 0.199852 \\
w_{14}(2) &= w_{14}(1) + \Delta w_{14} = 0.8 + (-0.00163) = -0.798368 \\
w_{24}(2) &= w_{24}(1) + \Delta w_{24} = 0.1 + (-0.00245) = 0.097552 \\
w_{04}(2) &= w_{04}(1) + \Delta w_{04} = (-0.20 + (-0.00082)) = -0.20082
\end{aligned}$$

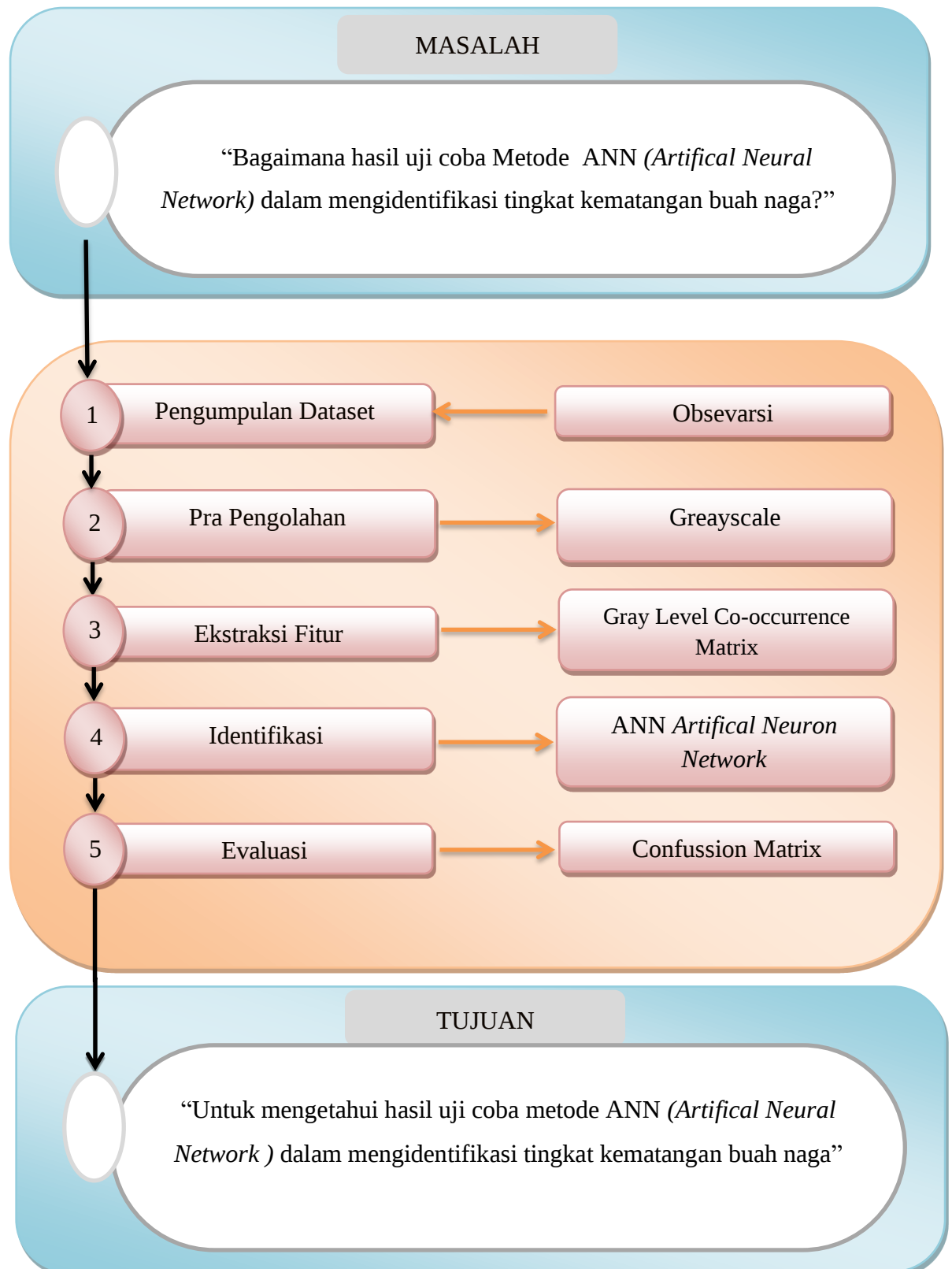
9. Naikkan 1 langkah iterasi (**p**), kembali ke langkah 2 dan ulangi proses tersebut sampai **kriteria error** tercapai.
10. Dengan demikian didapatkan bobot akhir pada iterasi pertama untuk data pertama [2.3]

$$\begin{aligned}
w_{13} &= 0.299704; \\
w_{23} &= 0.499556; \\
w_{03} &= 0.199852; \\
w_{14} &= -0.798368; \\
w_{24} &= 0.097552; \\
w_{04} &= -0.20082; \\
w_{35} &= 0.191887321; \\
w_{45} &= 0.692452718; \\
w_{05} &= 0.291073952
\end{aligned}$$

Error yang didapat adalah = **-0.886392478**. Selanjutnya proses di atas diulangi untuk data ke-dua [2, 6], data ke-tiga [1, 4], dan data ke-empat [1, 2] sehingga iterasi pertama selesai dilakukan. Pada setiap data yang diproses dalam satu iterasi, error keluaran disimpan untuk dihitung sebagai kriteria error, seperti SSE, MSE, dsb. Apabila hasil $MSE < \text{error yang didapatkan}$ maka iterasi berhenti, sebaliknya dilakukan perambatan terus hingga batas perulangan/epoch.

[13]

2. 3 Kerangka Pikir



BAB III

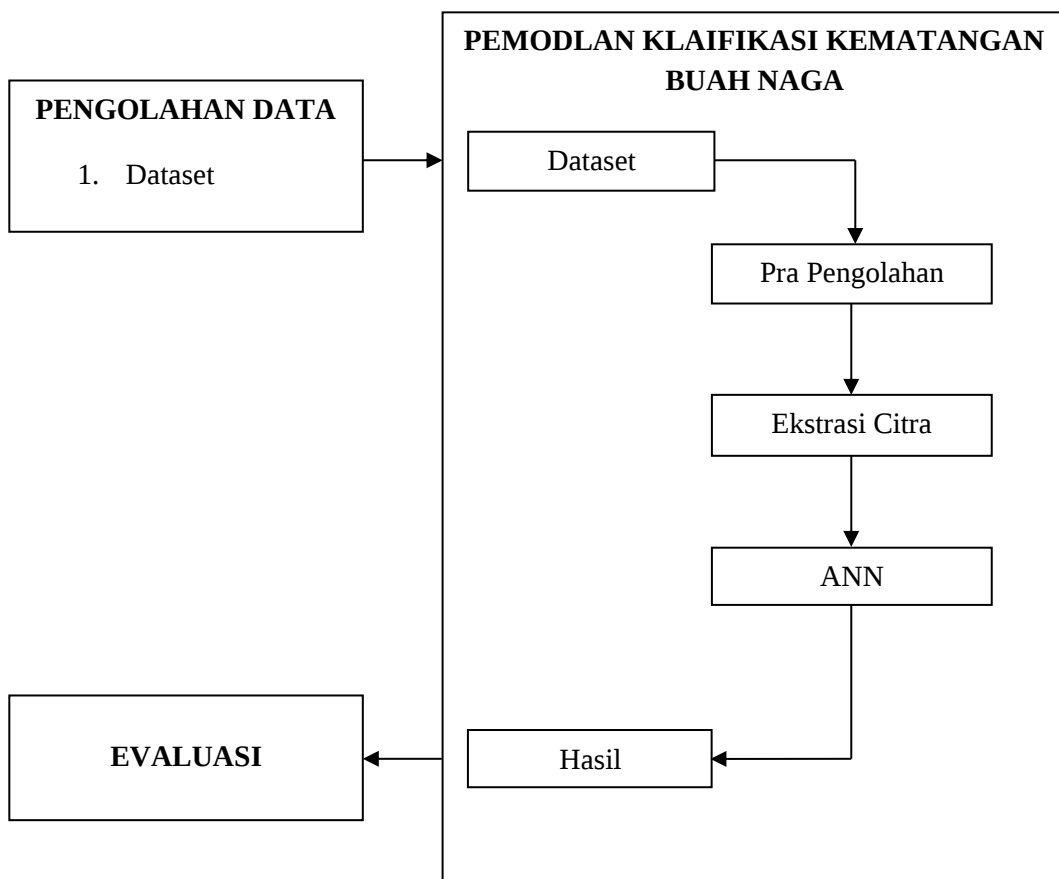
METODE PENELITIAN

3.1 Jenis, Metode, Subjel, Objek, dan Lokasi Penelitian

Dilihat dari tingkat penerapannya maka penelitian ini merupakan penelitian terapan. Dipandang dari jenis data yang diperoleh maka penelitian merupakan penelitian kuantitatif. Dipandang dari perlakuan terhadap data maka penelitian konfirmatori.

Penelitian ini menggunakan metode penelitian *eksperiment*. Subjek penelitian ini adalah *klasifikasi* pada objek buah naga.

Adapun metode penelitian ini diusulkan seperti gambar dibawah ini :



Gambar 3.1: Desain Penelitian

3.2 Pengumpulan Data

Data primer pada penelitian ini adalah buah naga yang dikumpulkan menggunakan teknik Obserfasi dan Dokumentasi. Dataset yang digunakan adalah dataset real, data yang akan digunakan dalam klasifikasi tingkat kematangan buah naga. Buah naga ini diambil di Des.Banurojo, Kec.Randangan Kab.Pohuato.

Jumlah gambar : 60 Gambar

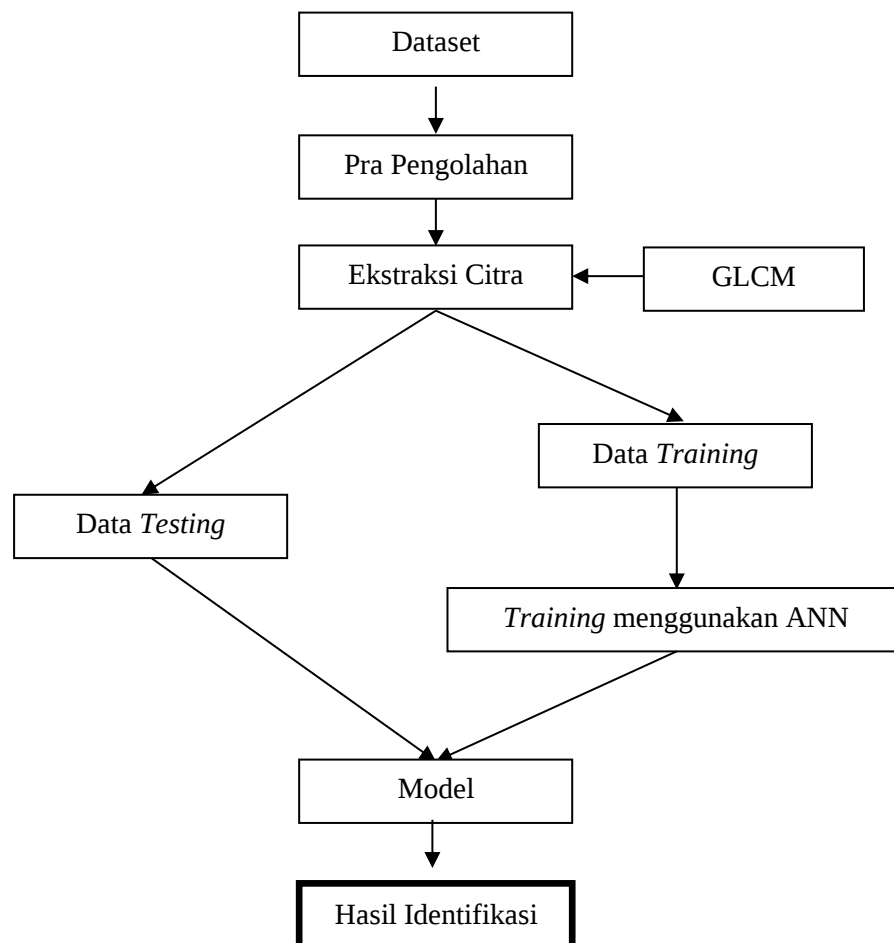
Tingkat Kematangan :Matang 99%, Matang 85%

Dokumentai : Kamera Handphone OPPO A5 2020

Resolusi : 12MP

Format : JPG

3.3 Pemodelan



Gambar 3.2: Pemodelan Identifikasi

3.3.1. Pra Pengolahan

Sebelum data masuk pengolahan, terlebih dahulu digunakan proses segmentasi, hal ini dilakukan untuk memisahkan antara objek yang dikehendaki.

3.3.2. Ekstrasi Fitur Warna

Ekstrasi fitur berfungsi sebagai pendeteksi ciri dari suatu citra. Ciri yang dapat digunakan untuk membedakan citra satu dengan citra yang lainnya, di antaranya adalah ciri bentuk, ciri ukuran, ciri geometri, ciri tekstur, dan ciri warna. Pada penelitian ini digunakan ekstraksi ciri warna menggunakan RGB (*Red, Green, Blue*). Masing-masing citra diekstraksi cirinya berdasarkan parameter-parameter tertentu dan dikelompokkan pada klas tertentu. Nilai dari parameter-parameter tersebut kemudian dijadikan sebagai data masukan dalam proses klasifikasi.

3.3.3. Data Training

Data *training* merupakan kumpulan data yang telah terekstraksi cirinya yang selanjutnya akan dilatih menggunakan algoritma *artificial neural network*. Algoritma ini akan menentukan atau mencari bobot yang terbaik. Data *training* berupa ciri dari hasil ekstraksi RGB (*Red, Green, Blue*) terdapat citra kematangan buah naga.

3.3.4. Training Menggunakan ANN

Training menggunakan ANN dilakukan dengan menjadikan data *Training* sebagai input untuk menentukan mode yang tepat. Arsitektur dari algoritma ANN akan dilakukan secara eksperiment untuk menentukan komposisi jumlah layer neuron yang guna mendapatkan hasil kinerja yang terbaik.

3.3.5. Model

model merupakan hasil dari proses *training* algoritma ANN menggunakan data *training*.

3.3.6. Data Testing

Data *testing* adalah sebuah data yang telah terekstraksi cirinya yang digunakan untuk menguji data yang telah dilatih. Data *testing* digunakan untuk mengukur sejauh mana penggolongan berhasil melakukan klasifikasi dengan benar. Maka dari itu, data yang ada pada data *testing* seharusnya tidak boleh ada pada data *training* sehingga dapat diketahui apakah model penggolongan sudah pintar dalam melakukan klasifikasi. Data *testing* berupa cirri dari hasil ekstraksi GLCM terhadap kematangan buah naga.

3.3.7. Hasil Klasifikasi

Hasil klasifikasi merupakan hasil *output* pada data *testing* yang didapat dari proses klasifikasi yang dilakukan oleh algoritma ANN berdasarkan model yang diperoleh dari hasil *training*.

3.4 Evaluasi

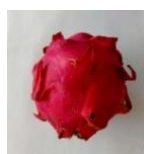
Evaluasi bertujuan untuk mengetahui kinerja dari metode tekstur analisis yang digunakan. Proses evaluasi dilakukan pada seluruh data *testing* kemudian target output yang dihasilkan akan dipetakan ke dalam *Confusion Matrix* untuk dihitung nilai akurasinya.

BAB IV HASIL PENELITIAN

4.1 Hasil Pengumpulan Data

Seperti yang telah dijelaskan pada tahapan metode penelitian bahwa dataset yang digunakan dalam penelitian ini merupakan data real. Peneliti telah melakukan proses pengumpulan data dan berhasil mnggumpulkan dengan jumlah data citra buah naga dengan tingkat kematangan yang berbeda sebanyak 60 data.

Tabel 4.1 Pengumpulan Data Set

Matang 99%				
Mentah 85%				

Untuk melanjutkan proses analisis pada penelitian ini maka dariitu kita harus melakukan pemilihan data gambar yang dijadikan sebagai data training dan data testing, jumlah data training adalah sebanyak 40 data gambar dan data testing adalah 20 gambar.

4.2 Hasil Pemodelan

4. 3. 1. PraPengolahan

Pra Pengolahan Citra (*image pre-processing*), yaitu proses paling awal dalam pengolahan citra sebelum proses utama dilakukan. Pada tahapan ini citra buah naga akan dikonversi untuk mendapatkan citra buah yang sesuai kebutuhan.

Proses awal yang dilakukan adalah merubah citra training atau citra testing yang awalnya citra dari RGB menjadi citra *grayscale*, perubahan ini dilakukan dikarenakan 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.



RumusKonversi Citra RGB ke *Grayscale*:

$$S = \frac{R + G + B}{3}$$

Dimana: R = Red, B = Blue, G = Green

Dimana:

Ko = Nilai keabuan hasil histogram

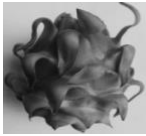
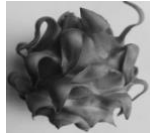
Ci = Distribusi komulatif dari nilai skala keabuan ke-I dari citra asli

Round = Fungsi pembulatan ke bilangan terdekat

W = Lebar Citra = Tinggi Citra

Pengubahan citra warna ke *grayscale* dan normalisasi citra dengan menggunakan *Histogram Equalization*, Setelah citraasli 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 agar setiap derajat keabuan memiliki jumlah piksel yang relative sama. Berikut proses perubahan citra warna ke *grayscale* dan normalisasi citra abu-abu dengan *histogram ekualisasi*:

Tabel 4.2 Hasil output dari hasil dari proses Konfersi RGB ke *Grayscale* dan *Grayscale* ke *Histogram Equalization*

Gambar Asli	Gambar Grayscale	Gambar Ekualisasi
		
		

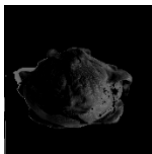
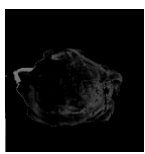
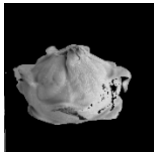
4. 3. 2. Citra *segmentasi*

Segmentasi citra adalah bagian dari proses pengolahan citra. Segmentasi citra (*image segmentation*) memiliki arti membagi suatu citra menjadi wilayah-wilayah yang homogeny 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 lebihlanjut.

4. 3. 3. Metode *Otsu Thresholding*

Metode otsu merupakan salah satu metode untuk segmentasi citra digital dengan menggnakan 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 Thresholding Selection Method from Grayscale Histogram*” pada tahun 1979.

Berikut ini proses citra RGB yang di ubah ke Segmentasi beserta perintah yang digunakan dalam bahasa *python* dan hasil outputnya:

Gambar Asili	Gambar R	Gambar G
		
Gambar B	Gambar Otsu	Gambar Segmentasi
		

Gambar 4.1 Citra Segmentasi

4. 3. 4. Ekstraksi Fitur

Tahap inidilakukan untuk proses mengubah gambar segmentasi menggunakan *ekstraksi fitur* GLCM. GLCM merepresentasikan hubungan dua piksel yang bertetangga dimana dua piksel yang saling berhubungan adalah yang 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° . Berikut contoh perhitungan *matrix* dengan ukuran $10 \times 10 \text{ pixel}$:

Tabel 4.3 Nilai dari Suatu Citra

r/k	0	1	2	3	4	5	6	7	8	9
0	176	176	175	175	174	174	177	177	177	177
1	175	175	175	175	175	175	176	176	177	177
2	174	175	175	175	175	175	176	176	176	176
3	174	175	175	175	175	175	176	176	176	176
4	174	174	175	175	175	175	175	176	176	176
5	173	174	174	175	175	175	175	176	176	176
6	173	173	174	176	175	175	175	175	175	175

r/k	0	1	2	3	4	5	6	7	8	9
7	174	174	175	176	175	174	175	175	175	175
8	175	174	175	175	175	174	175	175	175	175
9	175	175	175	175	174	174	175	175	175	176

Tabel 4.4 Matrix GLCM

x/y	173	174	175	176	177
173	1	2	0	0	0
174	2	4	13	0	1
175	0	13	39	9	0
176	0	0	9	12	1
177	0	1	0	1	4

Jumlah total nilai GLCM = 112

Selanjutnya pada tabel 4.6 nilai dalam pixel di jumlahkan dengan jumlah total dalam *pixel* sebelumnya.

Tabel 4.5 Matrix Ternormalisasi

x/y	173	174	175	176	177
173	0,0089	0,0178	0	0	0
174	0,0178	0,0347	0,1160	0	0,0089
175	0	0,0116	0,3482	0,0803	0
176	0	0	0,0803	0,1071	0,0089
177	0	0,0089	0	0,0089	0,0347

Setelah itu nilai *contras*, *homogeneity*, *energy* dan *entropy* di hitung secara manual di bawah ini :

a) Menghitung nilai *Contras*

***Contras* =**

$$\begin{aligned} & (173-173)^2 (0,0089) + (173-174)^2 (0,0178) + (174-173)^2 (0,0178) + (174-174)^2 \\ & (0,0347) + (174-175)^2 (0,1160) + (174-177)^2 (0,0089) + (175-174)^2 (0,0116) + \\ & (175-175)^2 (0,3482) + (175-176)^2 (0,0803) + (176-175)^2 (0,0803) + (176-176)^2 \\ & (0,1071) + (176-177)^2 (0,0089) + (177-174)^2 (0,0089) + (177-176)^2 (0,0089) + \\ & (177-177)^2 (0,0347) \end{aligned}$$

***Contras* =**

$$\begin{aligned} & (0*0,0089) + (1*0,0178) + (2*0,0178) + (0*0,0347) + (1*0,1160) + (9*0,0089) \\ & + (2*0,0116) + (0*0,3482) + (1*0,0803) + (2*0,0803) + (0*0,1071) + \\ & (1*0,0089) + (6*0,0089) + (2*0,0089) + (0*0,0347) \end{aligned}$$

***Contras* =**

$$\begin{aligned} & 0 + 0,0178 + 0,0356 + 0 + 0,1160 + 0,0801 + 0,0232 + 0 + 0,0803 + 0,1606 + \\ & 0 + 0,0089 + 0,0534 + 0,0178 + 0 \end{aligned}$$

$$\mathbf{Contras = 0,5937}$$

b) Menghitung nilai *Energy*

***Energy* =**

$$\begin{aligned} & (0,0089)^2 + (0,0178)^2 + (0,0178)^2 + (0,0347)^2 + (0,1160)^2 + (0,0089)^2 + \\ & (0,0116)^2 + (0,3482)^2 + (0,0803)^2 + (0,0803)^2 + (0,1071)^2 + (0,0089)^2 + \\ & (0,0089)^2 + (0,0089)^2 + (0,0347)^2 \end{aligned}$$

***Energy* =**

$$\begin{aligned} & 0,000079 + 0,000316 + 0,000316 + 0,001204 + 0,013456 + 0,000079 + \\ & 0,000134 + 0,121243 + 0,006448 + 0,006448 + 0,011470 + 0,000079 + \\ & 0,000079 + 0,000079 + 0,001204 \end{aligned}$$

$$\text{Energy} = 0,162634$$

c) Menghitung nilai Homogeneity

$$\text{Homogeneity} =$$

$$\begin{aligned} & ((0,0089) / (1 + (173-173))) + ((0,0178) / (1 + (173-174))) + ((0,0178) / (1 + \\ & (174-173))) + ((0,0347) / (1 + (174-174))) + ((0,1160) / (1 + (174-175))) + \\ & ((0,0089) / (1 + (174-177))) + ((0,0116) / (1 + (175-174))) + ((0,3482) / (1 + (175- \\ & 175))) + ((0,0803) / (1 + (175-176))) + ((0,0803) / (1 + (176-175))) + ((0,1071) / \\ & (1 + (176-176))) + ((0,0089) / (1 + (176-177))) + ((0,0089) / (1 + (177-174))) + \\ & ((0,0089) / (1 + (177-176))) + ((0,0347) / (1 + (177-177))) \end{aligned}$$

$$\text{Homogeneity} =$$

$$\begin{aligned} & 0,0089 + 0 + 0,0089 + 0,0347 + 0 + (-0,0045) + 0,0058 + 0,3482 + 0 + 0,0402 \\ & + 0,1071 + 0 + 0,0022 + 0,0045 + 0,0347 \end{aligned}$$

$$\text{Homogeneity} = 0,5907$$

d) Menghitung nilai Entropy

$$\text{Entropy} =$$

$$\begin{aligned} & (0,0089.\log 0,0089) + (0,0178.\log 0,0178) + (0,0178.\log 0,0178) + (0,0347.\log \\ & 0,0347) + (0,1160.\log 0,1160) + (0,0089.\log 0,0089) + (0,1160.\log 0,1160) + \\ & (0,3482.\log 0,3482) + (0,0803.\log 0,0803) + (0,0803.\log 0,0803) + (0,1071.\log \\ & 0,1071) + (0,0089.\log 0,0089) + (0,0089.\log 0,0089) + (0,0089.\log 0,0089) + \\ & (0,0347.\log 0,0347) \end{aligned}$$

$$\text{Entropy} =$$

$$\begin{aligned} & (0,0089*(-2,0506)) + (0,0178*(-1,7495)) + (0,0178*(-1,7495)) + (0,0347*(- \\ & 1,4596)) + (0,1160*(-0,9355)) + (0,0089*(-2,0506)) + (0,1160*(-0,9355)) + \\ & (0,3482*(-0,4581)) + (0,0803*(-1,0952)) + (0,0803*(-1,0952)) + (0,1071*(- \\ & 0,9702)) + (0,0089*(-2,0506)) + (0,0089*(-2,0506)) + (0,0089*(-2,0506)) + \\ & (0,0347*(-1,4596)) \end{aligned}$$

Entropy =

$$(-0,0182) + (-0,0311) + (-0,0311) + (-0,0506) + (-0,1085) + (-0,0182) + (-0,1085) + (-0,1595) + (-0,0879) + (-0,0879) + (-0,1039) + (-0,0182) + (-0,0182) + (-0,0182) + (-0,0506)$$

Entropy = -0,9104

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

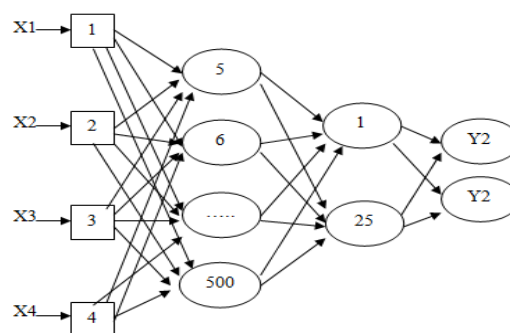
Ttabel 4.6 Hasil Nilai Fitur Ekstraksi

No	Contras	Energy	Homogeneity	Entropy
1	0,5937	0,162634	0,5907	-0,9104

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

Empat ciri yang di dapatkan dari *Matrix Co-Occurrence* inilah yang akan digunakan untuk tahap klasifikasi

4. 3. 5. Klasifikasi



Gambar 4.2 Arsitektur Jaringan Backpropagation

Klasifikasi pada data yang di gunakan yaitu 53 data Training dan 30 data Testing, yaitu terdiri dari 73 gambar buah naga matang dan 11 gambar buah naga mentah, dari 83 dataset yang telah di ambil, masing-masing gambar tersebut mempunyai nilai *contras*, *homogeneity*, *entropy*, dan *energy*. Seperti pada tabel 4.7 dibawah ini:

Tabel 4.7 Hasil Data Training

Data Training	Con (x1)	Ene (x2)	Hom (x3)	Ent (x4)	Y
	107.8044	0.3402	0.6553	0.9476	Matang 99%
	50.6657	0.5211	0.8628	0.9819	Matang 99%
	73.6496	0.3407	0.7329	0.9705	Matang 99%
	104.2028	0.3672	0.7217	0.9625	Matang 99%
	123.3798	0.4311	0.7240	0.9468	Matang 99%
	94.5004	0.3890	0.7354	0.9656	Matang 85%
	76.2630	0.5879	0.8137	0.9705	Matang 85%

Data Training	Con (x1)	Ene (x2)	Hom (x3)	Ent (x4)	Y
	74.4971	0.7780	0.8917	0.9706	Matang 85%
	84.5952	0.7391	0.8621	0.9711	Matang 85%
	74.4971	0.7780	0.8822	0.9725	Matang 85%

Tabel 4.8 Hasil Data Testing

Data Testing	Con (x1)	Ene (x2)	Hom (x3)	Ent (x4)	Y
 Matang 85%	75.8744	0.6696	0.8393	0.9829	Matang 85%
 Matang 85%	68.8290	0.8262	0.9095	0.9680	Matang 85%

 Matang 85%	44.4893	0.8330	0.9329	0.9730	Matang 85%
Data Testing	Con (x1)	Ene (x2)	Hom (x3)	Ent (x4)	Y
 Matang 85%	48.3818	0.7473	0.9045	0.9739	Matang 85%
 Matang 85%	55.8302	0.7617	0.8968	0.9743	Matang 85%
 Matang 99%	77.6835	0.3515	0.7467	0.9677	Matang 99%
 Matang 99%	67.0526	0.4706	0.7893	0.9771	Matang 99%
 Matang 99%	50.9806	0.5318	0.8102	0.9831	Matang 99%

 Matang 99%	63.4008	0.5778	0.8579	0.9751	Matang 99%
 Matang 99%	84.5952	0.8262	0.9095	0.9711	Matang 99%

Selanjutnya kita akan melakukan proses learning pada tabel data training diatas dengan menggunakan arsitektur jaringan MLP *backpropagation* yang telah disajikan sebelumnya.

Langkah awal yang dilakukan pada proses learning jaringan MLP tersebut adalah dengan menentukan nilai bobot (w), adapun nilai bobot awal yang dipilih yaitu sesuai dengan program dengan data test nomor 1 :

$$\begin{array}{llll}
 W_{15} = 0.3 & W_{25} = 0.2 & W_{35} = 0.1 & W_{45} = 0.3 \\
 W_{16} = 0.1 & W_{26} = 0.4 & W_{36} = 0.7 & W_{46} = 0.8 \\
 W_{17} = 0.5 & W_{27} = 0.7 & W_{37} = 0.4 & W_{47} = 0.2 \\
 W_{18} = 0.9 & W_{28} = 0.8 & W_{38} = 0.5 & W_{48} = 0.4 \\
 W_{51} = 0.6 & W_{61} = 0.3 & W_{71} = 0.7 & W_{81} = 0.4
 \end{array}$$

Tahap training menggunakan algoritma ANN:

Laju pembelajaran (η) = 0.1;

Fungsi aktivasi yang digunakan adalah Sigmoid Biner;

Target error = 0.0001 dengan kriteria SSE

Maksimum jumlah *iterasi* pelatihan = 1,000 kali.

Nilai pada *neuron* hidden layer 1,2, 3, , dan 500 :

untuk mendapatkan hasil actual output pada *neuron* 5 dan 6 dapat dihitung dengan seperti berikut :

$$:\gamma_5 = \text{sigmoid}(x_1w_{15} + x_2w_{25} + x_3w_{35} + x_4w_{45} \dots)$$

$$:\gamma_5 = 1/[1 + e^{-(75.8744 \times 0.3 + 0.6696 \times 0.2 + 0.8393 \times 0.1 + 0.9829 \times 0.3)}] = 0.1136$$

$$:\gamma_6 = \text{sigmoid}(x_1w_{16} + x_2w_{26} + x_3w_{36} + x_4w_{46} \dots)$$

$$:\gamma_6 = 1/[1 + e^{-(75.8744 \times 0.1 + 0.6696 \times 0.4 + 0.8393 \times 0.7 + 0.9829 \times 0.8)}] = 0.9999$$

$$:\gamma_7 = \text{sigmoid}(x_1w_{17} + x_2w_{27} + x_3w_{37} + x_4w_{47} \dots)$$

$$:\gamma_7 = 1/[1 + e^{-(75.8744 \times 0.5 + 0.6696 \times 0.7 + 0.8393 \times 0.4 + 0.9829 \times 0.2)}] = 0.4484$$

$$:\gamma_8 = \text{sigmoid}(x_1w_{18} + x_2w_{28} + x_3w_{38} + x_4w_{48} \dots)$$

$$:\gamma_8 = 1/[1 + e^{-(75.8744 \times 0.9 + 0.6696 \times 0.8 + 0.8393 \times 0.5 + 0.9829 \times 0.4)}] = 0.1417$$

Dengan diperolehnya *actual output* dari *neuron* 5, 6, 7 dan 8 maka nilai *actual output* dari *neuron* 1 pada layer 2 sudah dapat dihitung dengan menggunakan persamaan yang sama dengan yang digunakan pada *neuron* 5, 6, 7 dan 8 500

$$\gamma_1 = \text{sigmoid}(\gamma_5w_{51} + \gamma_6w_{61} + \gamma_7w_{71} + \gamma_8w_{81}) \dots$$

$$\gamma_1 = 1/[1 + e^{-(0.1136 \times 0.6 + 0.9999 \times 0.3 + 0.4484 \times 0.7 + 0.1417 \times 0.4)}] = 0.6766$$

Sinyal error dapat dihitung dengan menggunakan cara berikut :

$$e = \gamma_{d,1} - \gamma_1$$

$$e = 0 - 0.6766 = -0.6766$$

Setelah diperoleh nilai *error*, proses *training* dilakukan untuk proses update nilai bobot dan *level threshold* pada jaringan MLP, dapat dilakukan dengan cara propagasi nilai *error* (e) dari *output layer* yang secara *backward* hingga ke input layer.

Dalam melakukan propagasi balik, pertama-tama dilakukan kalkulasi nilai *gradient error* untuk *neuron 5* pada *output layer*, *gradient error* tersebut dihitung dengan cara berikut :

$$\delta_1 = \gamma_1(1 - \gamma_1)e$$

$$\delta_1 = 0.6766 \times (1 - 0.6766) \times (-0.6766) = -0.1480$$

Selanjutnya kita menentukan nilai koreksi bobot dengan asumsi learning rate (α) adalah sebesar 0.0001:

$$\nabla w_{51} = \alpha \times \gamma_5 \times \delta_1$$

$$\nabla w_{51} = 0,0001 \times 0,1136 \times (-0.1480) = -0.0000016$$

$$\nabla w_{61} = \alpha \times \gamma_6 \times \delta_1$$

$$\nabla w_{61} = 0.0001 \times 0.9999 \times (-0.1480) = -0.0000014$$

$$\nabla w_{71} = \alpha \times \gamma_7 \times \delta_1$$

$$\nabla w_{71} = 0.0001 \times 0.4484 \times (-0.1480) = -0.0000056$$

$$\nabla w_{81} = \alpha \times \gamma_8 \times \delta_1$$

$$\nabla w_{81} = 0.0001 \times 0.1417 \times (-0.1480) = -0.0000020$$

Selanjutnya hitung nilai *gradient error* untuk *neuron 5, 6, 7 dan 8.....* yang terdapat pada hidden layer:

$$\delta_5 = \gamma_5(1 - \gamma_5) \times \delta_1 \times w_{51}$$

$$\delta_5 = 0,1136 \times (1 - 0,1136) \times (-0.1480) \times 0.6 = -0.008941$$

$$\delta_6 = \gamma_6(1 - \gamma_6) \times \delta_1 \times w_{61}$$

$$\delta_6 = 0.9999 \times (1 - 0.9999) \times (-0.1480) \times 0.3 = -0.0000047$$

$$\delta_7 = \gamma_7(1 - \gamma_7) \times \delta_1 \times w_{71}$$

$$\delta_7 = 0.4484 \times (1 - 0.4484) \times (-0.1480) \times 0.7 = -0.0256$$

$$\delta_8 = \gamma_8(1 - \gamma_8) \times \delta_1 \times w_{81}$$

$$\delta_8 = 0.1417 \times (1 - 0.1417) \times (-0.1480) \times 0.4 = -0.0071$$

Maka nilai dari koreksi bobot dapat ditentukan sebagai berikut :

$$\Delta w_{15} = \alpha \times x_1 \times \delta_5$$

$$\Delta w_{15} = 0.0001 \times 75.8744 \times (-0.008941) = -0.00006783$$

$$\Delta w_{25} = \alpha \times x_2 \times \delta_5$$

$$\Delta w_{25} = 0.0001 \times 0.6696 \times (-0.008941) = -0.00000059$$

$$\Delta w_{35} = \alpha \times x_3 \times \delta_5$$

$$\Delta w_{35} = 0.0001 \times 0.8393 \times (-0.008941) = -0.00000074$$

$$\Delta w_{45} = \alpha \times x_4 \times \delta_5$$

$$\Delta w_{45} = 0.0001 \times 0.9829 \times (-0.008941) = -0.00000087$$

$$\Delta w_{16} = \alpha \times x_1 \times \delta_6$$

$$\Delta w_{16} = 0.0001 \times 75.8744 \times (-0.0000047) = -3.5658$$

$$\Delta w_{26} = \alpha \times x_2 \times \delta_6$$

$$\Delta w_{26} = 0.0001 \times 0.6696 \times (-0.0000047) = -3.1021$$

$$\Delta w_{36} = \alpha \times x_3 \times \delta_6$$

$$\Delta w_{36} = 0.0001 \times 0.8393 \times (-0.0000047) = -3.9011$$

$$\Delta w_{46} = \alpha \times x_4 \times \delta_6$$

$$\Delta w_{46} = 0.0001 \times 0.9829 \times (-0.0000047) = -4.6061$$

$$\Delta w_{17} = \alpha \times x_1 \times \delta_7$$

$$\Delta w_{17} = 0.0001 \times 75.8744 \times (-0.0256) = -0.00091$$

$$\Delta w_{27} = \alpha \times x_2 \times \delta_7$$

$$\Delta w_{27} = 0.0001 \times 0.6696 \times (-0.0256) = -0.0000016$$

$$\Delta w_{37} = \alpha \times x_3 \times \delta_{7s}$$

$$\Delta w_{37} = 0.0001 \times 0.8393 \times (-0.0256) = -0.0000021$$

$$\Delta w_{47} = \alpha \times x_4 \times \delta_7$$

$$\Delta w_{47} = 0.0001 \times 0.9829 \times (-0.0256) = -0.0000025$$

$$\Delta w_{18} = \alpha \times x_1 \times \delta_8$$

$$\Delta w_{18} = 0.0001 \times 75.8744 \times (-0.0071) = -0.000053$$

$$\Delta w_{28} = \alpha \times x_2 \times \delta_8$$

$$\Delta w_{28} = 0.0001 \times 0.6696 \times (-0.0071) = -0.00000046$$

$$\Delta w_{38} = \alpha \times x_3 \times \delta_8$$

$$\Delta w_{38} = 0.0001 \times 0.8393 \times (-0.0071) = -0.00000058$$

$$\Delta w_{48} = \alpha \times x_4 \times \delta_8$$

$$\Delta w_{48} = 0.0001 \times 0.9829 \times (-0.0071) = -0.00000069$$

Kemudian yang akan dilakukan update seluruh nilai bobot (w) dan *threshold* (θ) yang terdapat pada jaringan MLP, dengan menggunakan cara berikut:

$$w_{15} = w_{15} + \Delta w_{15} = 0.3 + (-0.00006783) = 0.2999$$

$$w_{16} = w_{16} + \Delta w_{16} = 0.1 + (-3.5658) = -3.4656$$

$$w_{17} = w_{17} + \Delta w_{17} = 0.5 + (-0.00091) = 0.4990$$

$$w_{18} = w_{18} + \Delta w_{18} = 0,9 + (-0.000053) = 0.8999$$

$$w_{25} = w_{25} + \Delta w_{25} = 0,2 + (-0.00000059) = 0.1999$$

$$w_{26} = w_{26} + \Delta w_{26} = 0.4 + (-3.1021) = 2,7021$$

$$w_{27} = w_{27} + \Delta w_{27} = 0.7 + (-0.0000016) = 0,6999$$

$$w_{28} = w_{28} + \Delta w_{28} = 0.8 + (-0.00000046) = 0.7999$$

$$w_{35} = w_{35} + \Delta w_1 = 0.1 + (-0.00000074) = 0.0999$$

$$w_{36} = w_{36} + \Delta w_{36} = 0.7 + (-3.9011) = -3.2011$$

$$w_{37} = w_{37} + \Delta w_{16} = 0.4 + (-0.0000021) = 0.3999$$

$$w_{38} = w_{38} + \Delta w_{16} = 0.5 + (-0.00000058) = 0.4999$$

$$w_{45} = w_{45} + \Delta w_{45} = 0.3 + (-0.00000087) = 0.2999$$

$$w_{46} = w_{46} + \Delta w_{46} = 0.8 + (-4.6061) = -3.8061$$

$$w_{47} = w_{47} + \Delta w_{47} = 0.2 + (-0.0000025) = 0.1999$$

$$w_{48} = w_{48} + \Delta w_{48} = 0.4 + (-0.00000069) = 0.3999$$

$$w_{1200}.....$$

Selanjutnya proses diatas diulangi sampai w_{1200} untuk hidden layer awal dan 10 untuk hidden layer 2 dan untuk data ke-dua data ke-tiga dan data ke-15. Pada setiap data yang diproses dalam satu iterasi, error keluaran disimpan untuk dihitung sebagai kriteria error, seperti SSE, MSE, dsb. Apabila hasil $MSE < \text{error}$ yang didapatkan maka iterasi berhenti, sebaliknya dilakukan perambatan terus hingga batas perulangan/*epoch*.

4.3 Evaluasi

4.3.6. *Confusion Matrix*

Pada penelitian ini menggunakan *Confusion Matrix* sebagai metode. Dalam perhitungan akurasi pada penerapan identifikasi tingkat kematangan buah naga. Evaluasi kinerja identifikasi kematangan di dasarkan dari jumlah pengujian objek salah dan benar yang di deteksi dapat dilihat dari table berikut :

Tabel 4.9 Hasil Confusion Matrix

		IDENTIFIKASI	
		Matang 85 %	Matang 99%
AKTUAL	Matang 85 %	5	0
	Matang 99 %	1	4

Jumlah data yang di identifikasi dengan benar = 9

Jumlah data yang di identifikasi salah = 1

$$\text{Akurasi} = 9/10 * 100\% = 90\%$$

BAB V

PEMBAHASAN

5. 1 Pembahasan Pengumpulan Data

Pengumpulan data yang diambil dari data real dengan jumlah data sebanyak 60 image, 60 image yang dipakai dalam penelitian ini terdiri dari 40 data training dan 20 data testing yang merupakan data image RGB dengan format JPG. Masing-masing data terdiri dari dua kategori, untuk data training terdiri dari 20 image buah naga matang 85% dan 20 image buah naga matang 99% dan. Sedangkan untuk data testing terdiri dari 10 image buah naga matang 85% dan 10 image buah naga matang 99%.

5. 2 Pembahasan Model

Sebelum pengolahan data dilakukan terlebih dahulu peneliti melakukan pra pengolahan data yaitu mengkonversi data training dan data testing dari image RGB ke Grayscale. Hal ini dilakukan karena inputan dari metode *Gray Level Co-Occurrence Matrix* merupakan inputan image grayscale. Maka seluruh data image RGB harus dikonversi ke grayscale. Setelah didapatkan hasil dari konversi citra, langkah selanjutnya yang dilakukan yaitu menormalisasi citra menggunakan Histogram Equalization. Proses selanjutnya yaitu pengolahan data, seluruh dataset (training dan testing) akan diekstraksi ciri menggunakan metode *Gray Level Co-Occurrence Matrix*. Ciri yang akan dihasilkan dari ekstraksi ciri yaitu contrast, homogenitas, entropy, dan energy. Hasil ekstraksi ciri inilah yang akan menjadi data inputan pada proses training menggunakan metode Artificial Neural Network, inputanya yaitu x_1 = Contrasts, x_2 = Energy, x_3 =Homogenitas, x_4 = Entropy.

Ada beberapa pelatihan data yang dilakukan untuk mengatur tingkat akurasi tertinggi. Pelatihan data pertama : menggunakan orientasi arah 0° , jarak piksel = 1, fungsi aktivasi = logistic, hidden layer 1 = 500 neuron, hidden layer 2 = 25 neuron

dan iterasi maksimum = 1000x, mampu menghasilkan tingkat akurasi yang dihitung menggunakan confusion matrix adalah 90%.

5.3 Pembahasan System

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

Gambar 5.1 :Tampilan Sistem

1. Input data gambar training

Gambar 5.2 Input gambar training

2. input data gambar testing

Gambar 5.3 Input gambar testing

3. Input nilai Gray Level Co-Occurrence Matrix (GLCM), terdiri dari :

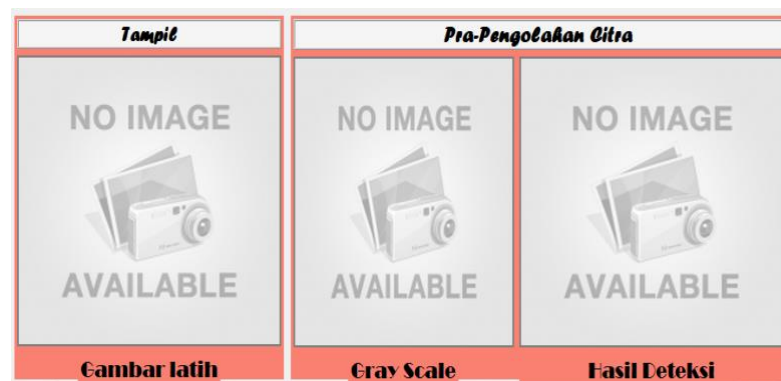
- Jarak keterangan: 1, 2, 3 dan seterusnya.
- Orientasi sudut arah: 0^0 , 45^0 , 90^0 , 135^0



Gambar 5.4 Input nilai

4. Prapengolahan

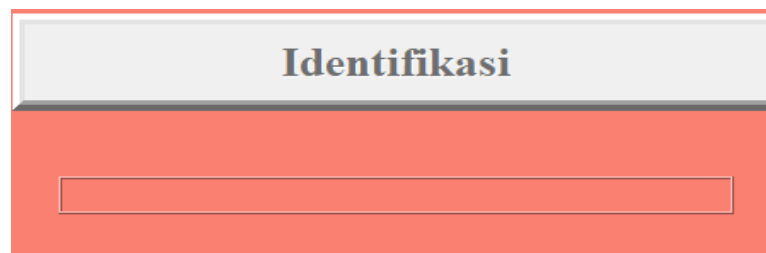
Pada prapengolahan akan menampilkan output berupa hasil konversi image dari RGB ke Grayscale, hasil normalisasi citra menggunakan Histogram Equalization, deteksi kematangan buah naga menggunakan Grayscale dan Segmentasi.



Gambar 5.5 Prapengolahan

1. Hasil klasifikasi

Hasil identifikasi menampilkan kategori tingkat kematangan buah naga mentah maupun matang.



Gambar 5.6 Hasil

BAB VI

PENUTUP

6. 1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan diatas maka dapat ditarik suatu kesimpulan bahwa hasil penerapan metode Gray Level Co-Occurrence Matrix (GLCM) untuk mengekstraksi ciri buah naga menggunakan metode Artificial Neural Network (ANN) mampu menghasilkan tingkat akurasi yang dihitung menggunakan counfusion matrix yaitu sebesar 90%

6. 2 Saran

Setelah melakukan penelitian, ada beberapa saran yang perlu diperhatikan untuk mencapai tujuan yang diharapkan, yaitu sebagai berikut :

1. Perlu diadakan penelitian lebih lanjut untuk mengidentifikasih buah naga, dengan penggunaan metode lainnya terlebih pada metode ekstraksi ciri sebagai perbandingan untuk mendapatkan hasil terbaik.
2. Untuk penelitian selanjutnya bisa menggunakan Algoritma lain sebagai perbandingan untuk mendapatkan hasil terbaik.

DAFTAR PUSTAKA

- [1] Resty Wulanningrum, Nandha Vera Wihra Lelitavistara “Discrete Cosine Transform Untuk Identifikasi Citra *Hylocereus Costaricensis*”, Jurnal SIMESTRIS, Vol 6, No 2, November 2015.

- [2] Neneng, Yusra Fernando “Klasifikasi Jenis Daging Berdasarkan Analisis Citra Tekstur Gray Level Co-Occurrence Matrices (GLCM) dan Warna” P-ISSN : 2407- 1846, E-ISSN : 2460- 8416, 2017.
- [3] Mochammad Yusuf Habibi, Edwin Riksakomaro “Peramalan Harga Garam Konsumsi Menggunakan Artificial Neural Network Feedforward-Backpropagation (Studi Kasus : PT. Garam Mas, Rembang, Jawa Tengah)”, Jurnal Teknik ITS, Vol 6, No 2, 2017.
- [4] Muh Najib Hilmi, Yuciana Wilandri, Hasbi Yasin “Pemetaan Preferensi Mahasiswa Baru Dalam Memilih Jurusan Menggunakan Artificial Neural Network (ANN) Dengan Algoritma Self Organizing Maps (SOM)”, Jurnal Gaussian, Vol 4, No 1, 2015
- [5] Andi Diah Kuswanto dan Hilman Pardede, “Komparasi Algoritma C4.5, K-Nearest Neighbor dan Neural Network Dalam Menentukan Kelayakan Fasilitas Kredit : Studi Kasus PT. Bank Mega, Tbk”, SNIPTEK, ISBN: 978-602-72850-4-0, 2013
- [6] Aulia Yudha Prathama, Akhmad Aminullah, Ashar Saputra, “Pendekatan ANN (*Artificial Neural Network*) Untuk Menentukan Prosentase Bobot Pekerjaan Dan Estimasi Nilai Pekerjaan Struktur Pada Rumah Sakit Prataa”, Jurnal Teknosains, Vol 7, No 1, Desember 2017.
- [7] Daniel Kristanto “Berkebun Buah Naga”, Cet. 1 (edisi revisi) Jakarta: Penebar Swadaya, ISBN (10) 979-002-629-3, ISBN (13) 978-979-002-629-2, 2014.
- [8] Warisno, S.PKP, Kres Dahana, SP “ Buku Pintar Bertanam Buah Naga Di Kebun, Pekarangan, Dan Dalam Pot”, Jakarta: PT.Gramedia Pustaka Utama, ISBN: 978-979-22-5404-4, 2010.
- [9] Kahirul Umam, Beni Sukma Negara, “Deteksi Obyek Manusia Pada Basis Data Video Menggunakan Metode *Background Subtraction* Dan Operasi Morfologi”, Jurnal CorIT, Vol 2, No 2, Desember 2016.

- [10] Tamrin Imanuel Panggabean, “Pendeteksi Dan Pengenalan Wajah Manusia Untuk Peningkatan kinerja Kamera Pengaman”, Skripsi Informatika, Februari 2018
- [11] Restu Widodo, Agus Wahyu Widodo, Arry Suprianto “Pemenfaatan Ciri *Gray Love Co-Occurrence Matrix* (GLCM) Citra Buah Jeruk Keprok (*Citrus Reticulata Blance*) Untuk Klasifikasi Mutu” Jurnal Pengembangan Teknologi Dan Ilmu Computer, Vol 2 No. 11, 2018.
- [12] Joko S. Dwi Raharjo, “Model Artificial Neural Network Berbasis Particle Swarm Optimization Untuk Pediksi Laju Inflasi”, Jurnal Sistem Komputer Vol 3. No 1, Juni 2013.
- [13] Q. Cahyo, “Sistem Cerdas Megenali Buah Jeruk Berkualitas Baik Menggunakan Metode *Artificial Neural Network*”, Skripsi, Teknik Informatika, Universitas Ichsan Gorontalo, Gorontalo, Indonesia 2017.

Lampiran 1: KODE PROGRAM

1. Pembuatan Form

```

import cv2

import tkinter as tk

from tkinter import filedialog

from PIL import ImageTk, Image

import tkinter.ttk as ttk

#import numpy as np

#import proses as p

from sklearn import preprocessing

import gray as proses_analisis

class Main(tk.Frame):

    def __init__(self, parent):

        tk.Frame.__init__(self)

        self.parent = parent

        self.parent.columnconfigure(3, weight=1)

        #frame judul

        self.MainFrame = tk.Frame(self.parent, background="white",
highlightthickness=8, bg="Salmon")

        self.MainFrame.pack(side="top", fill="x")

```

```

#frame judul

self.MainFrame0 = tk.Frame(self.parent, background="white",
highlightthickness=8, bg="Salmon")

self.MainFrame0.pack(side="top", fill="x")

#frame input gambar

self.MainFrame1 = tk.Frame(self.parent, highlightcolor="white",
highlightthickness=4, bg="Salmon")

self.MainFrame1.pack(side="top", fill="x")

#frame teks proses

self.MainFrame2 = tk.Frame(self.parent, highlightcolor="white",
highlightthickness=4, bg="Salmon")

self.MainFrame2.pack(side="top", fill="x")

#frame tampuil citra

self.MainFrame3 = tk.Frame(self.parent, highlightcolor="white",
highlightthickness=4, bg="Salmon")

self.MainFrame3.pack(side="left", fill="y")

#frame output

self.MainFrame4 = tk.Frame(self.parent, highlightcolor="white",
highlightthickness=4, bg="Salmon")

self.MainFrame4.pack(side="right", fill="y")

self.MainFrame5 = tk.Frame(self.parent, highlightcolor="white",
highlightthickness=4, bg="Salmon")

self.MainFrame5.columnconfigure(3, weight=1)

```

```

self.MainFrame5.pack(side="top", fill="both")

self.form()

def form (self):

#frame

    #Teks Judul Aplikasi

    self.PathTestingImg = "E:\\Program\\Logo1.png"

    self.imT = Image.open(self.PathTestingImg)

    self.imT = ImageTk.PhotoImage(self.imT)

    self.label1 = tk.Label(self.MainFrame, bg="Salmon", image=self.imT)

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

#frame 0

    #Teks Proses Perhitungan

    self.lblProses = tk.Label(self.MainFrame0, text="Masukkan Gambar",
font="forte 16 bold", bg="Salmon", anchor="center")

    self.lblProses.grid(padx=8, pady=10)

    self.lblProses.pack()

#frame 1

    #Label Images Latih

    self.label1 = tk.Label(self.MainFrame1, text="Gambar Latih",
font="Broadway 12", relief="ridge", background="Salmon")

    self.label1.grid(row=0, column=0, sticky="NSEW", padx=8, pady=10)

    #Teks Box Images Latih

```

```

bukaDataLatih = tk.StringVar( self.MainFrame1, value='')

self.textboxdatatraining = tk.Entry(self.MainFrame1,
textvariable=bukaDataLatih, width=148, bg='white smoke')

self.textboxdatatraining.grid(row=0, column=1, sticky="NSEW", padx=4,
pady=4)

#Tombol Browse Images Latih

self.path = tk.Button(self.MainFrame1, text="Pilih", font="Times 10 bold",
width=15, command=lambda:bukaDataLatih.set(filedialog.askdirectory()))

self.path.grid(row=0, column=2, sticky="NSEW", padx=3, pady=3)

#Label Images Testing

self.label2 = tk.Label(self.MainFrame4, text="Gambar Uji",
font="Broadway 12", relief="ridge", background="Salmon")

self.label2.grid(row=0, column=0, sticky="NSEW", padx=4, pady=4)

#Teks Box Images Testing

bukaDataTest = tk.StringVar(self.MainFrame4, value='')

self.textboxdatatesting = tk.Entry(self.MainFrame4,
textvariable=bukaDataTest, width=55, bg='white smoke')

self.textboxdatatesting.grid(row=0, column=1, sticky="NSEW", padx=4,
pady=4)

#Tombol Browse Images Testing

self.path1 = tk.Button(self.MainFrame4, text="Pilih", font="Times 10 bold",
width=15,

state='disable',

command=lambda:bukaDataTest.set(filedialog.askopenfilename()))

```

```

self.path1.grid(row=1, column=1, sticky="NSEW", padx=3, pady=3)

#Tombol Proses

self.tbltrain = tk.Button(self.MainFrame1, text="Training", font="Times 15
bold", width=10, command=self.Training)

self.tbltrain.grid(row=0, column=3, rowspan=2, sticky="NSEW", padx=3,
pady=3)

self.tbltrain.configure(borderwidth=10, relief="raised", anchor="center")

#frame 2

#Teks Proses Perhitungan

self.lblProses = tk.Label(self.MainFrame2, text="Proses Identifikasi",
font="forte 16 bold", bg="Salmon", anchor="center")

self.lblProses.grid(padx=3, pady=3)

self.lblProses.pack()

#frame 3

#proses prapengolahan

self.lblTmpl = tk.Label(self.MainFrame3, text="Tampil", font="forte 14
bold", relief="ridge", background="white smoke")

self.lblTmpl.grid(row=0, column=0, columnspan=2, padx=3, pady=3,
sticky="NSEW")

self.asli = 'fon.png'

self.noimageAsli = Image.open(self.asli)

self.noimageAsli = self.noimageAsli.resize((300, 260), Image.ANTIALIAS)

self.noimageAsli = ImageTk.PhotoImage(self.noimageAsli)

```

```

self.CitraAsli = tk.Label(self.MainFrame3, bg="gray",
image=self.noimageAsli)

self.CitraAsli.grid(row=1, column=0, padx=3, pady=3, sticky="NSEW")

self.Citrateks = tk.Label(self.MainFrame3, text="Gambar latih",
font="Broadway 16", background="Salmon")

self.Citrateks.grid(row=2, column=0, padx=3, pady=3)

#frame 5

self.lblPra = tk.Label( self.MainFrame5, text="Pra-Pengolahan Citra",
font="forte 16 bold", relief="ridge", background="white smoke")

self.lblPra.grid(row=0, column=0, columnspan=2, padx=3, pady=3,
sticky="NSEW")

self.asli = 'fon2.png'

self.noimageGray = Image.open(self.asli)

self.noimageGray = self.noimageGray.resize((250,260),
Image.ANTIALIAS)

self.noimageGray = ImageTk.PhotoImage(self.noimageGray)

self.CitraGS = tk.Label( self.MainFrame5, bg="gray",
image=self.noimageGray)

self.CitraGS.grid(row=1, columnspan=1, padx=3, pady=3)

self.CitraGSteks = tk.Label( self.MainFrame5, text="Gray Scale",
font="Broadway 16 ", background="Salmon")

self.CitraGSteks.grid(row=2, column=0, padx=3, pady=3)

self.asli = 'fon3.png'

```

```

self.noimageHist = Image.open(self.asli)

self.noimageHist = self.noimageHist.resize((300,260), Image.ANTIALIAS)

self.noimageHist = ImageTk.PhotoImage(self.noimageHist)

self.CitraHist = tk.Label( self.MainFrame5, bg="gray",
image=self.noimageHist)

self.CitraHist.grid(row=1, column=1, padx=3, pady=3)

self.CitraHisteks = tk.Label( self.MainFrame5, text="Hasil Deteksi",
font="Broadway 16 ", background="Salmon")

self.CitraHisteks.grid(row=2, column=1, padx=3, pady=3, sticky="NSEW")

#frame 4

self.label1 = tk.Label(self.MainFrame1, text="Jarak :", font="Broadway 12",
relief="ridge", background="Salmon")

self.label1.grid(row=1, column=0, sticky="NSEW", padx=1, pady=3)

self.label1 = tk.Label(self.MainFrame1, text="Arah :", font="Broadway 12",
relief="ridge", background="Salmon")

self.label1.grid(row=2, column=0, sticky="NSEW", padx=4, pady=4)

#Teks Box Images Latih Jarak GLCM

self.jarakglcm = tk.StringVar(self.MainFrame1, value="")

self.tbjarak = ttk.Combobox(self.MainFrame1, textvariable=self.jarakglcm ,
width=150, state='readonly')

self.tbjarak['values']=( '1', '2', '3', '4', '5')

self.tbjarak.current(0)

self.tbjarak.grid(row=1, column=1, padx=4, pady=4, sticky="E")

```

```

#Teks Box Images Arah GLCM

self.arahglcm = tk.StringVar(self.MainFrame1, value="")

self.tbarah = ttk.Combobox(self.MainFrame1, textvariable=self.arahglcm,
width=150, state='readonly')

self.tbarah['values']=( '0', '45', '90', '135')

self.tbarah.current(0)

self.tbarah.grid(row=2, column=1, padx=4, pady=4, sticky="E")

#Hasil program

self.tblProses = tk.Button(self.MainFrame4, text="Identifikasi",
                             font="Times 20 bold", width=15, state='disable',
                             command=self.testter)

self.tblProses.grid(row=5, columnspan=3, rowspan=1, sticky="NSEW",
padx=3, pady=3)

self.tblProses.configure(borderwidth=7, relief="raised", anchor="center")

#Label Images Hasil Identifikasi

self.label4 = tk.Label(self.MainFrame4, text="", width=20, relief='ridge',
font="Broadway 12", background="Salmon")

self.label4.grid(row=6, column=0, columnspan=2, sticky="NSEW",
padx=30, pady=40)

def Training(self):

    folder_train = self.textboxdatatraining.get()

    self.jarak = self.tbjarak.get()

```

```

self.arah = self.tbarah.get()

self.rekognisi = proses_analisis.image_analisis()

self.model_ann, self.model_dataEncoder, self.model_label =
self.rekognisi.model_algoritma(folder_train, self.jarak, self.arah)

self.tblProses.configure(state='active')

self.path1.configure(state='active')

def tester(self):

    image_test = self.textboxdatatesting.get()

    x_test, img_test, img_pra, segmen_test =
self.rekognisi.testing_data(image_test, self.jarak, self.arah)
    x_test = self.model_dataEncoder.transform(x_test)

    hasil = self.model_ann.predict(x_test)

    hasil = self.model_label.inverse_transform(hasil)

    self.label4.configure(text=hasil[0])

    #menampilkan gambar

    im_asli = cv2.cvtColor(img_test, cv2.COLOR_BGR2RGBA)

    im_asli = Image.fromarray(im_asli)

    im_asli = im_asli.resize((300, 260), Image.ANTIALIAS)

    img_asli = ImageTk.PhotoImage(im_asli)

    self.CitraAsli.configure(image=img_asli)

    self.CitraAsli.image_names = img_asl

    #menampilkan gambar

```

```

im_gray = Image.fromarray(img_pra)

im_gray = im_gray.resize((300, 260), Image.ANTIALIAS)

img_gray = ImageTk.PhotoImage(im_gray)

self.CitraGS.configure(image=img_gray)

self.CitraGS.image_names = img_gray

#menampilkan gambar

im_hist = cv2.cvtColor(segmen_test, cv2.COLOR_BGR2RGBA)

im_hist = Image.fromarray(im_hist)

im_hist = im_hist.resize((300, 260), Image.ANTIALIAS)

img_hist = ImageTk.PhotoImage(im_hist)

self.CitraHist.configure(image=img_hist)

self.CitraHist.imamge_names = img_hist

def main():

    root = tk.Tk()

    root.geometry('1030x650+160+20')

    root.configure(background='Salmon')

    root.state("zoomed")

    root.resizable(1,1)

    app = Main(root)

    app.parent.title("Identifikasi Kematangn Buah Naga")

```

```

root.mainloop()

if __name__ == '__main__':

    main()

```

2. Proses

```

import cv2
from imutils import paths
import numpy as np
#fromb scipy.spatial import convexHull
from skimage import feature
from scipy.stats import entropy
from scipy import ndimage
import os
from scipy import stats
from sklearn.neural_network import MLPClassifier as mlp
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import LabelEncoder
import warnings
with warnings.catch_warnings():
    warnings.filterwarnings("ignore", category=DeprecationWarning)
class image_analisis:
    def get_path(self, folder_train):
        array_path = []
        for path in paths.list_images(folder_train):
            array_path. Append(path)
        return array_path

    def getLabelTrain(self, folder_train):
        label = []
        path = self.get_path(folder_train)

```

```

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

def pra_pengolahan(self, im_path):
    img = cv2.imread(im_path)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    kerner = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]])
    result = cv2.filter2D(gray, -1, kerner)
    return result

def segmentasi(self, img):
    img = cv2.imread(img)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    blur = cv2.GaussianBlur(gray,(5,5),0)
    ret, alpha = cv2.threshold(blur, 70, 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, img

def glcmfitur(self, img, neigh, angle):
    glcm = feature.greycomatrix(img, [neigh], [angle], symmetric=False,
                                normed=True)
    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 [con, ene, hmg, cor]

def training_data(self, folder_training, distance, angle):
    feature_train = []
    im_path = self.get_path(folder_training)
    for image in im_path:
#         hasil_prapengolahan = self.pra_pengolahan(image)
        hasil_segmentasi, gray, img = self.segmentasi(image)
        gray_segmentasi = cv2.cvtColor(hasil_segmentasi,
cv2.COLOR_BGR2GRAY)
        hasil_glcm = self.glcmfitur(gray_segmentasi, distance, angle)
        feature_train.append(hasil_glcm)
    return feature_train

def testing_data(self, img_test, distance, angle):
    feature_test = []
#     hasil_prapengolahan = self.pra_pengolahan(img_test)
    hasil_segmentasi, gray, img = self.segmentasi(img_test)
    gray_segmentasi = cv2.cvtColor(hasil_segmentasi,
cv2.COLOR_BGR2GRAY)
    hasil_glcm = self.glcmfitur(gray_segmentasi, distance, angle)
    feature_test.append(hasil_glcm)
    return feature_test, img, gray, hasil_segmentasi
def dataEncoder(self, data):
    scaler = MinMaxScaler(feature_range=(0, 1))
    scaler.fit(data)
    return scaler
def label_encoder(self, label):
    model_label = LabelEncoder()
    model_label.fit(label)

```

```

    result = model_label.transform(label)
    return model_label, result

def model_algoritma(self, folder_train, distance, angle):
    x_training = self.training_data(folder_train, distance, angle)
    model_dataEncoder = self.dataEncoder(x_training)
    x_training = model_dataEncoder.transform(x_training)
    y_training = self.getLabelTrain(folder_train)
    model_label, y_training = self.label_encoder(y_training)
    #x_test = self.testing_data(img_test, distance, angle)
    model_ann = mlp(activation = 'logistic',
                    hidden_layer_sizes=([500,25]),
                    max_iter=1000,
                    learning_rate_init=0.0001,
                    solver='lbfgs')
    model_ann.fit(x_training, y_training)
    return model_ann, model_dataEncoder, model_label

```

Lampiran 2 : RIWAYAT HIDUP PENELITI

A. Biodata

- | | |
|-------------------------|--|
| 1. Nama | : Siti Nurkholiza Anwar |
| 2. Jenis Kelamin | : Perempuan |
| 3. Tempat Tanggal Lahir | : Atinggola 18 November 1997 |
| 4. Kebangsaan | : Indonesia |
| 5. Agama | : Islam |
| 6. Email | : sitinanwar@gmail.com |



B. Riwayat Pendidikan

- | | |
|--------|---|
| 1. TK | : TK Puspita Bintana 2003-2004 |
| 2. SD | : SD Negri 1 Bintana 2004-2010 |
| 3. SMP | : SMP Negri 1 Atinggola 2010-2013 |
| 4. SMA | : SMA Negri 3 Gorontalo Utara 2013-2016 |

C. Pengalaman Organisasi

1. Bendahara Umum Persatuan Pelajar Mahasiswa Atinggola (P2MA) 2019-2021
2. Pengurus Pergerakan Mahasiswa Islam Indonesia (PMII) Rayon Fikom 2017-2019
3. Wakil Sekertaris Karang Taruna SIGASO Desa Sigaso 2021-2024

Lampiran 3 : SURAT REKOMENDASI PENELITIAN



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS IHSAN GORONTALO

FAKULTAS ILMU KOMPUTER

SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001

Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT KETERANGAN PENELITIAN

Nomor : 497/FIKOM-UIG/SKP/XII/2020

Yang bertanda tangan dibawah ini :

N a m a : Zohrahayaty, M. Kom
 Jabatan : Dekan Fakultas Ilmu Komputer

Dengan ini Menerangkan bahwa :

N a m a : Siti Nurkholiza Anawar
 N I M : T3116022
 Program Studi : Teknik Informatika

Bahwa yang bersangkutan benar-benar telah melakukan penelitian tentang "**Tingkat Kematangan Buah Naga Menggunakan Metode Artificial Neural Network (ANN)**", dan bersangkutan telah menyelesaikan penelitian pada **TGL 30 November 2020**.

Demikian Surat Keterangan ini dibuat untuk digunakan seperlunya.

Gorontalo, 07 Desember 2020
 Dekan,

Zohrahayaty, M.Kom
 NIDN : 0912117702

Lampiran 4: BUKTI PENERIMAAN SOFTCOPY SKRIPSI



Pustikom
Universitas Ichsan Gorontalo

BUKTI PENERIMAAN SOFTCOPY SKRIPSI PENGECEKAN SIMILARITY TURNITIN

Nama Mahasiswa : SITI NURKHOLIZA ANWAR
NIM : T3116022
Program Studi : Teknik Informatika (S1)
Fakultas : Fakultas Ilmu Komputer
Judul Skripsi : Identifikasi Tingkat Kematangan Buah Naga Menggunakan Metode ANN

Nama File (Pdf) : HASIL_SKRIPSI_T3116022_SITI NURKHOLIZA ANWAR

No. HP/WA : 082362609303

e-Mail : Sitinonwar@gmail.com

Tgl. Terima : 11/12/2020

--	--	--	--	--	--

Hasil Pengecekan : 16%

--	--	--	--	--	--

Diterima/Diperiksa Oleh,


Sudirman S. Panna, M.Kom
085340910769

Lampiran 5: SURAT REKOMENDASI BEBAS PLAGIASI



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS ICHSAN
(UNISAN) GORONTALO
SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT REKOMENDASI BEBAS PLAGIASI

No. 0787/UNISAN-G/S-BP/XII/2020

Yang bertanda tangan di bawah ini:

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

Dengan ini Menyatakan bahwa :

Nama Mahasisw : SITI NURKHOLIZA ANWAR
NIM : T3116022
Program Studi : Teknik Informatika (S1)
Fakultas : Fakultas Ilmu Komputer
Judul Skripsi : Identifikasi Tingkat Kematangan Buah Naga Menggunakan Metode ANN

Sesuai dengan hasil pengecekan tingkat kemiripan skripsi melalui aplikasi Turnitin untuk judul skripsi di atas diperoleh hasil Similarity sebesar 16%, berdasarkan SK Rektor No. 237/UNISAN-G/SK/IX/2019 tentang Panduan Pencegahan dan Penanggulangan Plagiarisme, bahwa batas kemiripan skripsi maksimal 35% dan sesuai dengan Surat Pernyataan dari kedua Pembimbing yang bersangkutan menyatakan bahwa isi softcopy skripsi yang diolah di Turnitin SAMA ISINYA dengan Skripsi Aslinya serta format penulisannya sudah sesuai dengan Buku Panduan Penulisan Skripsi, untuk itu skripsi tersebut di atas dinyatakan BEBAS PLAGIASI dan layak untuk diujikan.

Demikian surat rekomendasi ini dibuat untuk digunakan sebagaimana mestinya.

Gorontalo, 12 Desember 2020

Tim Verifikasi,



Sunarto Taliki, M.Kom
NIDN. 0906058301

Tembusan :

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

Lampiran 6: HASIL TURNITIN



Siti NurKholiza Anwar (T3116022).docx
Dec 11, 2020
9305 words / 61928 characters

T3116022 Siti NurKholiza Anwar

Identifikasi Kematangan Buah Naga Menggunakan Metode ANN

Sources Overview

16%

OVERALL SIMILARITY

1	www.scribd.com	2%
2	eprints.undip.ac.id	2%
3	jurnal.umk.ac.id	2%
4	j-pstik.ub.ac.id	2%
5	repository.upi.edu	1%
6	medianeliti.com	<1%
7	ejurnal.its.ac.id	<1%
8	jurnal.umj.ac.id	<1%
9	eprints.dnus.ac.id	<1%
10	ejournal.itn.ac.id	<1%
11	repository.ub.ac.id	<1%
12	id.123dbk.com	<1%
13	digilib.uinsgd.ac.id	<1%
14	eprints.unianu.ac.id	<1%
15	eprints.umg.ac.id	<1%
16	repository.usu.ac.id	<1%

17	repository.its.ac.id	INTERNET	<1%
18	jurnal.unsi.ac.id	INTERNET	<1%
19	edoc.pub	INTERNET	<1%
20	www.pythontic	INTERNET	<1%
21	jurnal.iain.ac.id	INTERNET	<1%
22	textid123dok.com	INTERNET	<1%
23	ojs.amikom.ac.id	INTERNET	<1%

Excluded search repositories:

- Submitted Works

Excluded from Similarity Report:

- Small Matches (less than 25 words)

Excluded sources:

- None