

**KLASIFIKASI KUALITAS DAGING SAPI
MENGUNAKAN METODE *K-NEAREST
NEIGHBOR* DENGAN *EKSTRAKSI
FITUR GRAY LEVEL CO-
OCCURRENCE MATRIX***

Oleh

TRISNO

T3113205

SKRIPSI

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



**PROGRAM SARJANA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2021**

PENGESAHAN SKRIPSI

**KLASIFIKASI KUALITAS DAGING SAPI
MENGUNAKAN METODE K-NEAREST
NEIGHBOR DENGAN EKSTRAKSI
FITUR GRAY LEVEL CO-
OCCURRENCE MATRIX**

Oleh
Trisno
T313205

SKRIPSI

Untuk Memenuhi Salah Satu Syarat Ujian Guna Memperoleh
Gelar Sarjana Pada Program Studi Teknik Informatika.
Ini Telah Disetujui Oleh Tim Pembimbing

Gorontalo, 22 Juni 2021

Pembimbing Utama


Yasin Arif Mustofa, M.Kom
NIDN : 0926088503

Pembimbing Pendamping


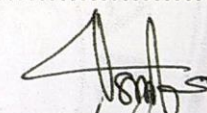
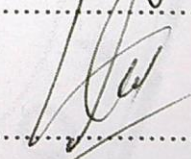
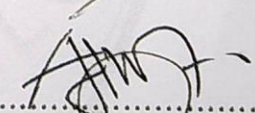
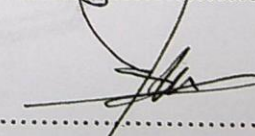

Sunarto Taliki, M.Kom
NIDN : 0906058301

PERSETUJUAN SKRIPSI
KLASIFIKASI KUALITAS DAGING SAPI
MENGGUNAKAN METODE K-NEAREST
NEIGHBOR DENGAN EKSTRAKSI
FITUR GRAY LEVEL CO-
OCCURRENCE MATRIX

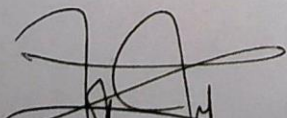
Oleh
Trisno
T3113205

Diperiksa oleh Panitia Ujian Strata (S1)
Universitas Ichsan Gorontalo
Gorontalo, 22 Juni 2021

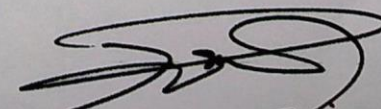
1. Ketua Penguji
H. Amiruddin, M.Kom
2. Anggota
Asmaul Husna, M.Kom
3. Anggota
Husdi, M.Kom
4. Anggota
Yasin Aril Mustofa, M.Kom
5. Anggota
Sunarto Taliki, M.Kom


:

:

:

:

:

Dekan Fakultas Ilmu Komputer


Zohrahavaty, M.Kom
NIDN : 0912117702

Ketua Program Studi


Irvan Abraham Salihi, M.Kom
NIDN : 0928028101

REDMI NOTE 8
AI QUAD CAMERA



HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa :

1. Karya tulis saya (skripsi) 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 ini adalah murni gagasan, rumusan dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari Tim Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah dipublikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini 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 yang berlaku di perguruan tinggi ini.

Gorontalo, 01 Juni 2021

Yang Membuat Pernyataan,

TRISNO

T3113205

ABSTRACT

***TRISNO. T3113205. THE QUALITY CLASSIFICATION OF BEEF USING
K-NEAREST NEIGHBOR METHOD WITH GRAY LEVEL CO-
OCCURRENCE MATRIX FEATURE EXTRACTION***

*This study aims to create a system that is able to overcome difficulties in the process of recognizing beef quality and knowing the results of applying the KNN method with GLCM feature extraction for the procurement process to recognize beef quality. This research uses the descriptive research method, which is a method that describes a situation that is currently taking events at the time the research is conducted and examines the causes of a certain symptom systematically based on the existing data. In this study, the system testing phase will be carried out by testing 20 test image data with the *.jpg extension. From the 20 test image data, the parameter values will be investigated by selecting the direction and distance and determined based on 00, 450, 900, and 1350. Thus, based on the calculation of the closest distance to the test data using $K = 3$, then the classification results are obtained with categories Fresh quality and accuracy calculation results obtained from fresh and not fresh beef through the Confusion Matrix calculation which produces an accuracy rate of 0.85% (85%).*

Keywords: Implementation Results, Feature Extraction, KNN Method, GLCM

ABSTRAK

TRISNO. T3113205. KLASIFIKASI KUALITAS DAGING SAPI MENGGUNAKAN METODE K-NEAREST NEIGHBOR DENGAN EKSTRAKSI FITUR GRAY LEVEL CO-OCCURRENCE MATRIX

Penelitian ini bertujuan untuk membuat sebuah sistem yang mampu mengatasi kesulitan dalam proses mengenali kualitas daging sapi dan mengetahui hasil penerapan metode KNN dengan ekstraksi fitur GLCM untuk proses pengadaaan mengenali kualitas daging sapi. Dalam penelitian ini menggunakan metode penelitian deskriptif, yaitu suatu metode yang menggambarkan suatu keadaan yang sementara berjalan pada saat penelitian dilakukan dan memeriksa sebab-sebab dari suatu gejala tertentu secara sistematis berdasarkan data-data yang ada. Dalam Penelitian ini tahap pengujian sistem akan dilakukan dengan melakukan pengujian terhadap 20 data citra uji yang ber ekstensi *.jpg. Dari 20 data citra uji tersebut akan dicari nilai parameternya dengan memilih arah dan jarak, di tentukan berdasarkan 0^0 , 45^0 , 90^0 , dan 135^0 . Dengan demikian, berdasarkan perhitungan jarak terdekat pada data uji menggunakan $K=3$, maka diperoleh hasil klasifikasi dengan kategori berkualitas Segar dan hasil perhitungan akurasi yang diperoleh dari daging sapi segar dan tidak segar melalui perhitungan Confusion Matrix di dapat tingkat akurasi sebesar 0,85% (85%).

Kata Kunci :Hasil Penerapan, Ekstraksi Fitur, Metode *KNN*, *GLCM*.

DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN JUDUL	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
HALAMAN PERNYATAAN	v
ABSTRACT	vi
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	4
1.3 Rumusan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
BAB II LANDASAN TEORI	6
2.1 Tinjauan Studi	6
2.2 Tinjauan Teori	8
2.2.1 Daging Sapi	8
2.2.2 Kecerdasan Buatan.....	10
2.2.3 Komputer Visi	12
2.2.4 Citra Digital	13
2.2.5 Pengolahan Citra Digital	19
2.2.6 Analisis	34
2.2.7 Perancangan	35
2.2.8 Implementasi	43

2.2.9	Pengujian.....	43
2.2.10	Pemeliharaan	45
2.3	Teknik Pengujian Sistem	45
2.3.1	<i>White Box</i>	45
2.3.2	<i>Black Box</i>	49
2.4	Pengujian Model	51
2.5	Perangkat Lunak Pengembang	53
2.6	Kerangka Pemikiran	55
BAB III OBJEK DAN METODE PENELITIAN		56
3.1	Objek Penelitian	56
3.2	Metode Penelitian	56
3.2.1	Tahap Pengumpulan Data	57
3.2.2	Tahap Analisis Sistem.....	57
3.2.2.1	Analisis Sistem Berjalan	58
3.2.2.2	Analisis Sistem Usulan.....	58
3.2.3	Tahap Desain Sistem	59
3.2.4	Tahap Pengujian Sistem	59
3.2.5	Tahap Pemeliharaan Sistem	60
3.2.6	Tahap Implementasi.....	60
BAB IV HASIL PENELITIAN		61
4.1	Hasil Pengumpulan Data	61
4.2	Arsitektur Metode KNN.....	62
4.3	Penerapan Metode	63
4.4	Nilai GLCM	65
4.5	Klasifikasi KNN	70
4.6	Sistem Yang Diusulkan	77
4.7	Activity Diagram.....	77
4.8	Sequence Diagram Pilih File.....	79
4.9	Sequence Diagram Proses GLCM.....	79
4.10	Sequence Diagram Data Uji	80
4.11	Arsitektur Sistem Klasifikas	

4.12	Desain Interface	81
BAB V PEMBAHASAN PENELITIAN		82
5.1	Hasil Penelitian	82
5.1.1	Gambaran Umum Lokasi Penelitian	82
5.1.1.1	Dinas Pertanian dan Peternakan.....	82
5.1.1.2	Tugas dan Fungsi.....	83
5.1.2	Hasil Pengujian Sistem.....	84
5.2	Pembahasan Sistem	86
BAB VI KESIMPULAN DAN SARAN		90
6.1	Kesimpulan.....	90
6.2	Saran.....	90
DAFTAR PUSTAKA		
LAMPIRAN		

DAFTAR GAMBAR

Gambar 2.1	Konsep Kecerdasan Buatan	11
Gambar 2.2	Sistem Visi Komputer	13
Gambar 2.3	Ilustrasi Citra Gambar	14
Gambar 2.4	Citra Biner.....	15
Gambar 2.5	Citra Skala Keabuan (gray scale).....	16
Gambar 2.6	Citra Warna (true color).....	17
Gambar 2.7	Hasil Mengubah citra RGB ke Citra Grayscale.....	23
Gambar 2.8	Siklus Hidup Pengembangan Sistem.....	34
Gambar 2.9	Bagan Alir.....	46
Gambar 2.10	Grafik Alir.....	47
Gambar 2.11	Antarmuka Program Aplikasi Pengembangan Anaconda	54
Gambar 2.12	Kerangka Pemikiran.....	55
Gambar 4.1	Arsitektur	62
Gambar 4.2	Pembagian Data Training Segar	63
Gambar 4.3	Pembagian Data Training Tidak Segar	64
Gambar 4.4	Pembagian Data Testing Segar	65
Gambar 4.5	Pembagian Data Testing Tidak Segar	65
Gambar 4.6	Sistem Yang Di usulkan.....	77
Gambar 4.7	Activity Diagram Menu File.....	77
Gambar 4.8	Activity Diagram Menu Proses	78
Gambar 4.9	Activity Diagram Testing.....	78
Gambar 4.10	Sequence Diagram Pilih File.....	79

Gambar 4.11	Sequence Diagram Proses GLCM.....	79
Gambar 4.12	Sequence Diagram Data Uji.....	80
Gambar 4.13	Desain Inteface Sistem.....	81
Gambar 5.1	Hasil Pengujian Sistem.....	86
Gambar 5.2	Tampilan Menu Utama.....	86
Gambar 5.3	Tampilan Load Gambar.....	87
Gambar 5.4	Tampilan Menentukan Arah dan Jarak.....	88
Gambar 5.5	Tampilan Menu Hasil Identifikasi.....	88

DAFTAR TABEL

Tabel 2.1 Matrix Co-Occurrence dari Matrix Asal	27
Tabel 2.3 GLCM simetris ternormalisasi dari matriks asal	27
Tabel 2.4 Contoh Kasus.....	32
Tabel 2.5 Contoh Hasil Klasifikasi.....	32
Tabel 2.6 Hasil Akhir.....	33
Tabel 2.7 Daftar pengelompokan diagram pada UML.....	38
Tabel 2.8 Notasi pengklasifikasian dalam UML.....	40
Tabel 2.9 Notasi relasi – relasi dalam UML.....	42
Tabel 2.10 Confusion Matrix.....	51
Tabel 2.11 Klasifikasi Akurasi.....	53
Tabel 4.1 Data Penelitian	61
Tabel 4.2 Nilai GLCM 0^0	66
Tabel 4.3 Nilai GLCM 45^0	67
Tabel 4.4 Nilai GLCM 90^0	68
Tabel 4.5 Nilai GLCM 135^0	69
Tabel 4.6 Data Gambar.....	70
Tabel 4.7 Menghitung kedekatan.....	72
Tabel 4.8 Berdasarkan urutan Ranking Jarak.....	73
Tabel 4.9 Kategori Antar Jarak.....	73
Tabel 4.10 Confusion Matrix.....	74
Tabel 4.11 Tingkat Ketepatan.....	75

Tabel 4.12 Tingkat Keberhasilan.....	75
Tabel 4.13 Tingkat Akurasi	76

BAB I

PENDAHULUAN

1.1. Latar belakang

Daging sapi salah satu jenis makanan mentah yang di gemari masyarakat luas di seluruh belahan dunia termasuk Indonesia. Banyaknya kandungan gizi pada daging sapi tentunya bermanfaat bagi tubuh manusia apabila daging sapi di konsumsi. Konsumsi daging sapi diperkirakan akan meningkat tiap tahunnya dan akan terus meningkat seiring dengan membaiknya perekonomian masyarakat.

Kab. Bone Bolango merupakan kabupaten termuda di Provinsi Gorontalo yang memiliki jumlah penduduk terbanyak ke dua, banyaknya penduduk berimbas terhadap permintaan daging sapi. Banyaknya permintaan daging sapi mengakibatkan berjamurnya penjual daging sapi, selain di pasar mingguan dan harian banyak pedagang daging sapi membuka lapak jualannya di tepi jalan. Meningkatnya permintaan tersebut memberikan celah kepada kelompok tertentu yang memanfaatkan situasi untuk meraup keuntungan yang lebih besar. Salah satunya adalah dengan cara menjual daging sapi gelongongan, yaitu sapi yang diberi minum secara paksa sebelum disembelih. Hal ini mengakibatkan bertambahnya kadar air dalam daging, sehingga daging sapi gelongongan menjadi lebih berat bila dibanding dengan daging sapi segar yang normal.

Daging sapi busuk atau daging gelongongan ini biasanya muncul pada hari – hari lebaran dan pada rentan harga yang biasanya melonjak naik. Pada kejadian yang seperti ini biasanya dimanfaatkan oleh para pedagang nakal yang

lebih mementingkan keuntungan semata tanpa memperhatikan kualitas daging itu sendiri. Fenomena seperti ini di dasari dengan adanya permintaan pasar yang semakin naik dan harga yang relatif tinggi. Banyak pedagang ingin mendapatkan keuntungan yang besar dengan modal yang kecil.

Dinas Peternakan Kab. Bone Bolango merupakan instansi yang bertanggung jawab dalam mengontrol persebaran penjualan daging sapi yang ada di wilayah Kab. Bone Bolango, dalam beberapa sidak ke beberapa lapak pedagang daging sapi sering menemukan adanya pedangan yang menjual daging sapi gelonggongan dan daging sapi yang tidak layak jual. Selama ini evaluasi kualitas dan identifikasi daging dilakukan oleh instansi terkait masih secara manual yaitu masih melalui pengamatan visual. Identifikasi dengan cara ini memiliki kelemahan, antara lain membutuhkan waktu yang lama dan menghasilkan produk dengan kualitas yang tidak konsiten karena keterbatasan visual manusia, kelelahan dan adanya perbedaan persepsi tentang kualitas pada masing-masing pengamat. Sehingga dibutuhkan adanya sistem yang dapat membantu dinas terkait dalam hal mengindentifikasi kualitas daging sapi, dengan adanya sistem ini dapat mempermudah dan mempersingkat waktu dalam mengindentifikasi daging sapi.

Pengolahan citra merupakan alternatif untuk persoalan ini. Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik atau segar. Pengenalan citra merupakan suatu mekanisme untuk mengenali kembali citra yang secara signifikan oleh mata tidak dapat dikenali lagi, namun dengan metode dan teknik tertentu citra tersebut masih dapat dikenali.

k-nearest neighbor (k-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Klasifikasi menggunakan metode k-nearest neighbor bertujuan mengklasifikasikan citra daging segar dan tidak segar.

Berdasarkan uraian diatas, maka dianggap perlu untuk melakukan penelitian mengenai proses yang berjalan diatas, dengan judul **“Klasifikasi Kualitas Daging Sapi Menggunakan Metode *K-Nearest Neighbor* Dengan Ekstraksi Fitur *Gray Level Co-Occurrence Matrix* (Studi Kasus : Pada Dinas Peternakan Kab. Bone Bolango)”**.

1.2. Identifikasi Masalah

Berdasarkan uraian yang telah diuraikan diatas, kemudian disesuaikan dengan kenyataan di lapangan maka dapat diidentifikasi permasalahannya sebagai berikut:

1. Belum efisiennya proses penentuan kualitas dan identifikasi daging sapi yang masih dilakukan dengan penilaian subjektif masing kurang efektif.
2. Belum ada aplikasi yang mengenali kualitas daging sapi menggunakan metode KNN dengan ekstraksi fitur GLCM.

1.3. Rumusan Masalah

Dari latar belakang diatas, maka dapat dirumuskan masalah:

1. Bagaimana cara merancang aplikasi yang berfungsi sebagai alat bantu dalam hal mengenali daging sapi yang berkualitas segar menggunakan metode KNN dengan ekstraksi fitur GLCM?

2. Bagaimana hasil akurasi metode KNN dengan ekstraksi fitur GLCM untuk proses mengenali kualitas daging sapi?

1.4. Tujuan Penelitian

Adapun tujuan dari pembuatan perangkat lunak ini antara lain adalah :

1. Merancang aplikasi klasifikasi kualitas daging sapi menggunakan metode KNN dengan ekstraksi fitur GLCM sebagai alat bantu mengenali kualitas daging sapi
2. Mengetahui hasil akurasi metode KNN dengan ekstraksi fitur GLCM dalam mengenali kualitas daging sapi

1.5. Manfaat Penelitian

Penelitian ini diharapkan mempunyai manfaat yaitu:

1. Pengembangan ilmu.

Penelitian ini diharapkan dapat mengembangkan ilmu pengetahuan dibidang teknologi komputer pada umumnya dan proses mengenali kualitas daging sapi khususnya.

2. Praktisi.

Sebagai bahan masukan (*Input Source*) bagi semua elemen-elemen ataupun unsur-unsur yang terlibat dalam pembuatan Sistem Klasifikasi Kualitas Daging Sapi Menggunakan Metode KNN dengan ekstraksi fitur GLCM.

3. Peneliti.

Sebagai masukan bagi peneliti lain yang akan mengadakan penelitian selanjutnya dan dapat memberikan informasi bagi mereka tentang masalah yang diteliti untuk menerapkannya dalam sistem yang lebih luas dan lebih kompleks.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Studi

Ada berapa penelitian tentang penggunaan metode KNN dengan ekstraksi fitur GLCM dalam aplikasi pengolahan citra adalah sebagai berikut :

1. Penelitian yang dilakukan oleh Hasanah, Uswatun(2018) tentang “Klasifikasi Citra Daging Sapi Menggunakan Metode KNN Untuk Identifikasi Kualitas Dan Kesegaran” Tujuan penelitian ini untuk mengetahui kualitas daging sapi yang segar dan daging busuk menggunakan co-occurrence matriks untuk mengklasifikasikan citra daging dengan perhitungan K-NN. Penelitian ini dapat dimanfaatkan untuk dapat membedakan jenis daging berdasarkan warna dan tekstur. Data yang digunakan dalam penelitian ini sebanyak 60 data citra yang terdiri dari 30 citra daging segar dan 30 citra daging busuk. Proses klasifikasi menggunakan data uji dengan tujuan dapat membedakan jenis daging segar dan daging busuk. Hasil identifikasi jenis daging berdasarkan warna dan tekstur untuk daging busuk dengan $K=3$ mendapatkan nilai sebesar 0,80638. Dari 60 jenis citra tersebut sistem menghasilkan 42 jenis data citra yang sesuai dan 18 jenis data citra yang tidak sesuai dengan objek citra yang sebenarnya. Dengan demikian sistem yang dibangun dengan metode K-NN cukup baik digunakan dengan tingkat keakuratan data yang didapat sebesar 70% dan mendapatkan data tidak akurat sebesar 30%.

2. Penelitian yang dilakukan oleh Elvia Budianita, Jasril, Lestari Handayani (2015) tentang “Implentasi Pengolahan Citra Dan Klasifikasi K-Nearest Neighbor Untuk Membangun Aplikasi Pembeda Daging Sapi Dan Daging Babi” . Pada penelitian ini akan dibuat suatu sistem pengolahan citra untuk membedakan daging sapi dan babi menggunakan metode HSV, GLCM, dan klasifikasi K-Nearest Neighbour (KNN). Tujuannya adalah untuk mengetahui hasil akurasi metode tersebut dalam klasifikasi citra daging sapi dan babi. Tahapan analisa yang dilakukan adalah Data acquisition dengan menggunakan kamera handphone Samsung Grand Two 8 MP dilakukan pemotretan terhadap data daging sapi dan babi sehingga diperoleh citra digital daging sapi dan babi dalam format jpg. Gambar (citra) diambil dari daging babi segar, daging sapi segar, dan daging sapi yang telah membusuk. Ekstraksi fitur menggunakan histogram model warna HSV untuk fitur warna, dan metode orde dua untuk ekstraksi fitur tekstur. Klasifikasi citra daging sapi dan babi menggunakan K-NN dengan dua tahapan yaitu tahap training dan testing. Pengujian dilakukan dengan menggunakan jumlah data latih dan nilai k yang berbeda. Hasil pengujian terhadap citra daging sapi dan babi tersebut memiliki akurasi tertinggi terletak pada pengujian tanpa background yaitu 88,75%, sedangkan pengujian dengan background memiliki akurasi keberhasilan 73,375%.
3. Penelitian yang dilakukan oleh Mungki Astininingrum, Mustika Mentari, Rezida Rismawati Nur Rachma (2019) tentang “Deteksi Kesegaran Daging Sapi Berdasarkan Ekstraksi Fitur Warna Dan Tekstur” Dalam penelitian ini akan dikembangkan suatu sistem yang dapat mendeteksi /mengklasifikasikan

kesegaran daging sapi berdasarkan fitur tekstur dan warna menggunakan metode Probabilistic Neural Network (PNN). Metode pendekatan statistik dan metode Gray Level Co-Occurance Matrix (GLCM) digunakan dalam proses ekstraksi fitur. Total dataset yang digunakan dalam penelitian ini adalah 80 dataset yang dibagi menjadi data latih dan data uji. Akurasi tertinggi yang diperoleh yaitu 75% menggunakan pasangan citra HSI &GLCM.

2.2. Tinjauan Teori

2.2.1 Daging Sapi

Daging sapi adalah daging yang diperoleh dari sapi yang biasa dan umum digunakan untuk keperluan konsumsi makanan. Daging sapi segar mengandung air 75%, protein 19%, dan lemak 2.5% (Syamsir, 2008). Komposisi daging menurut Direktorat Gizi Departemen Kesehatan RI (1981) dalam Soputan (2004) dalam 100 gram daging mengandung protein sebesar 18,8 gram dan lemak 14 gram. Daging mempunyai kandungan mineral antara lain kalsium 11 mg, fosfor 170 mg, dan besi 2,8 mg. Selain itu daging juga memiliki kandungan vitamin A dan vitamin B1. Ada beberapa faktor yang dapat dijadikan pedoman untuk memilih daging sapi segar antara lain :

a. Warna

Warna daging adalah salah satu kriteria penilaian mutu daging yang dapat dinilai langsung. Warna daging ditentukan oleh kandungan dan keadaan pigmen daging yang disebut mioglobin dan dipengaruhi oleh jenis hewan, umur hewan, pakan, aktivitas otot, penanganan daging dan reaksi-reaksi kimiawi yang terjadi di dalam daging. Warna daging sapi segar yang baik

adalah warna merah cerah. Warna daging sapi yang baru dipotong yang belum terkena udara adalah warna merah-keunguan, lalu jika telah terkena udara selama kurang lebih 15-30 menit akan berubah menjadi warna merah cerah. Warna merah cerah tersebut akan berubah menjadi merah-coklat atau coklat jika daging dibiarkan lama terkena udara.

b. Bau

Bau daging segar tidak berbau masam/busuk, tetapi berbau khas daging segar. Bau daging dipengaruhi oleh jenis hewan, pakan, umur daging, jenis kelamin, lemak, lama waktu, dan kondisi penyimpanan. Bau daging dari hewan yang tua relatif lebih kuat dibandingkan hewan muda, demikian pula daging dari hewan jantan memiliki bau yang lebih kuat daripada hewan betina. Kebusukan akan kerusakan daging ditandai oleh terbentuknya senyawa-senyawa berbau busuk seperti amonia, H_2S , indol, dan amin, yang merupakan hasil pemecahan protein oleh mikroorganisme

c. Tekstur

Daging segar bertekstur kenyal, padat dan tidak kaku, bila ditekan dengan tangan, bekas pijatan kembali ke bentuk semula. Daging yang tidak baik ditandai dengan tekstur yang lunak dan bila ditekan mudah hancur.

d. Kenampakan

Daging segar tidak berlendir, tidak terasa lengket ditangan dan terasa kebasahannya. Daging yang busuk sebaliknya berlendir dan terasa lengket di

tangan. Selain itu permukaan daging berwarna kusam, kotor dan terdapat noda merah, hitam, biru, putih kehijauan akibat kegiatan mikroba.

2.2.2 Kecerdasan Buatan

2.2.2.1 Definisi kecerdasan buatan

McCarthy mendefinisikan kecerdasan buatan atau *Artificial Intelligence* (AI) sebagai berikut. (Artificial Intelegent. Konsep dan penerapannya. Widodo Budiharto, Derwin Suhartono, 2014):

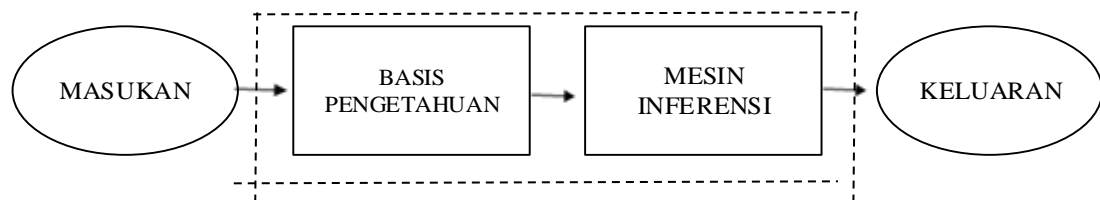
“AI merupakan cabang dari ilmu komputer yang berfokus pada pengembangan komputer untuk dapat memiliki kemampuan dan berperilaku seperti manusia.”

Manusia pandai dalam menyelesaikan permasalahan Karena manusia mempunyai pengetahuan dan pengalaman agar computer bertindak seperti sebaik manusia, maka komputer juga harus di bekali bekal pengetahuan dan mempunyai kemampuan untuk menalar.

Dua bagian utama konsep kecerdasan buatan, yaitu (Artificial Intellegence. Konsep dan penerapannya Widodo Budiharto, Derwin Suhartono, 2014):

1. Basis pengetahuan. Merupakan represetasi pengetahuan yang diperlukan untuk memahami memformulasikan, dan memecahkan masaalah. Terdiri dari dua elemen dasar yaitu:

- a) Fakta berupa informasi tentang situasi permasalahan, teori dari area permasalahan atau informasi tentang objek.
 - b) *Spatial heuristic*, merupakan informasi tentang cara membangkitkan fakta.
2. Mesin Inferensi. Membuat *Inferensi*. yang memutuskan aturan – aturan mana yang akan digunakan.



Gambar 2.1. Konsep Kecerdasan Buatan

2.2.2.2 Ruang Lingkup Kecerdasan Buatan

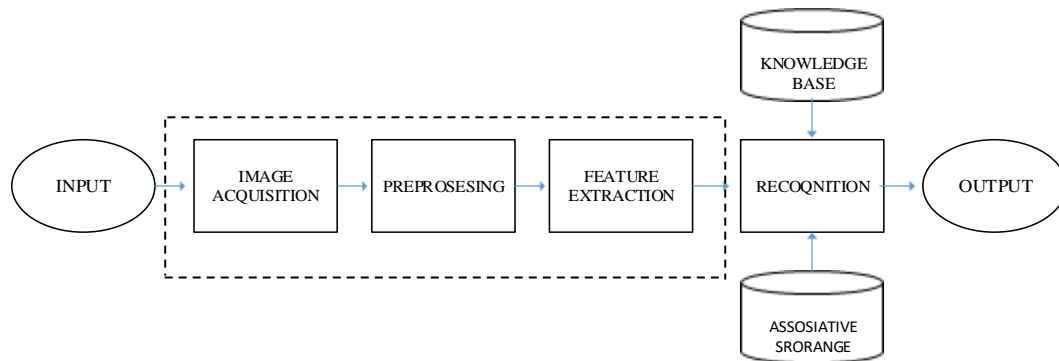
Adanya irisan penggunaan kecerdasan buatan di berbagai disiplin ilmu menyebabkan sulitnya pengklasifikasian berdasarkan disiplin ilmu. Oleh Karena itu, pengklasifikasian kecerdasan buatan dibuat berdasarkan keluaran yang dihasilkan. Ruang lingkup kecerdasan buatan adalah :

1. Sistem pakar (*expert system*) system pakar menggunakan penalaran dengan meniru atau mengadopsi keahlian yang dimiliki oleh pakar sehingga komputer dapat menyelesaikan permasalahan layaknya seseorang pakar.
2. Pengolahan Bahasa alami (*natural language processing*). Merupakan proses komputer yang melakukan ekstraksi informasi dari input yang

3. berupa natural language dan /atau menghasilkan output yang juga berupa natural language.
4. Robotika dan sistem navigasi bidang ilmu yang mempelajari bagaimana membuat sistem robot yang dapat menyelesaikan tugas - tugas nya
5. Visi Komputer (*computer vision*). Cabang ilmu yang erat kaitannya dengan pembangunan arti atau makna dari image ke objek fisik. Hal yang di butuhkan adalah metode untuk memperoleh, melakukan proses menganalisa, dan memahami image.
6. Permainan (*game playing*). Konsep kecerdasan buatan dapat di terapkan pada permainan. Tujuan intinya adalah membuat *non player* memiliki strategi yang cerdas untuk mengalahkan player.

2.2.3 Komputer Visi

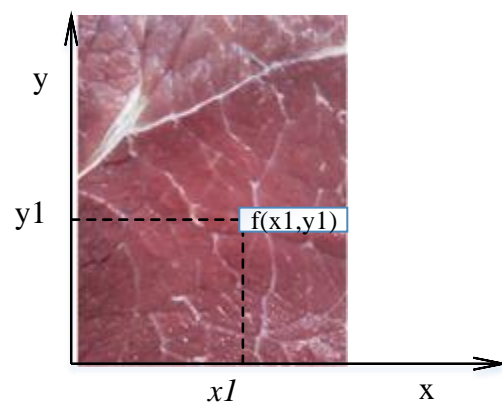
Salah satu pendekatan untuk mengimplementasikan sistem visi computer adalah dengan berusaha meniru sistem visi manusia. Namun, permasalahan yang ada pada pendekatan ini adalah sistem visi manusia sangat kompleks dan sulit di mengerti. Tidaklah mungkin untuk dapat meniru sistem visi manusia secara sempurna, walaupun demikian studi terhadap sistem biologis memberikan petunjuk untuk membangun sistem visi komputer tentunya ini bukanlah hal yang mustahil.



Gambar 2.2 Sistem Visi Komputer

2.2.4 Citra Digital

Citra adalah representasi informasi dua dimensi yang di ciptakan atau dibuat dengan melihat atau lebih tepatnya merasakan sebuah gambar atau pemandangan (Tabtras T., Onno W. Purbo, 2000). Dalam tinjauan matematis, citra merupakan fungsi kontinyu dari intensitas cahaya pada bidang dua dimensi. Ketika sumber cahaya menerangi objek, objek memantulkan kembali cahaya tersebut. Pantulan ini ditangkap oleh alat pengindra optik. Misalnya mata manusia, kamera, scanner, dan sebagainya. Bayangan objek tersebut akan terekam sesuai itensitas pantulan cahaya. Ketika alat optik yang merekam pantulan cahaya itu merupakan mesin digital, misalnya kamera digital, maka citra yang dihasilkan merupakan citra digital (Murni, 2004).



Gambar 2.3 Ilustrasi Citra Gambar

Sebuah citra digital dapat di anggap sebagai sebuah larik besar yang berisi sample point dari sebuah gambar kontinyu yang masing – masing point mempunyai kualitas kecerahan. Pengertian tersebut lebih di kenal dengan istilah piksel yang sering digunakan pada citra digital.

Piksel mempunyai dua parameter, yaitu koordinat, dan intentitas atau warna, nilai yang terdapat pada koordinat (x,y) adalah $f(x,y)$, yaitu besar intensitas atau warna dari titik piksel itu. Oleh sebab itu, sebuah citra digital dapat ditulis dalam bentuk matriks berikut.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,m-1) \\ f(1,0) & \cdots & \cdots & f(1,m-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(n-1,0) & f(n-1,1) & \cdots & f(n-1,m-1) \end{bmatrix}$$

Berdasarkan gambaran tersebut, secara matematis citra digital dapat dituliskan sebagai fungsi $f(x,y)$, dimana harga x (baris) dan y (kolom) merupakan koordinat posisi dan $f(x,y)$ adalah nilai fungsi pada setiap titik (x,y) yang menyatakan besar intensitas citra atau tingkat keabuan atau warna dari pixel di titik tersebut (sutoyo, et al, 2009).

Suatu piksel memiliki nilai dalam rentang tertentu, dari nilai minimum hingga maksimum. Jangkauan yang digunakan berbeda - beda tergantung dari jenis warnanya. Namun secara umum jangkauannya adalah 0 (nol) sampai dengan 255. Citra dengan penggambaran seperti ini di golongan kedalam citra integer (Putra, 2010). Citra digital secara umum dikelompokan menjadi 3 macam, yaitu :

- Citra Biner

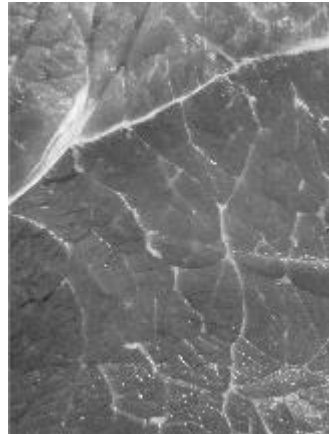
Dimana masing – masing pixel hanya berwarna hitam dan putih, maka hanya di perlukan satu bit per – pixel. Citra yang di representasikan sebagai citra biner sangat cocok digunakan untuk citra biner sangat cocok digunakan untuk citra yang menggambarkan text, finger print, atau rencana arsitektural.



Gambar 2.4. Citra Biner

- Citra abu – abu (Grayscale).

Dimana masing – masing pixel berisikan warna abu – abu dengan nilai normal antara 0 (hitam) sampai 255 (putih). Range tersebut berarti masing – masing pixel dapat direpresentasikan oleh nilai 8 bit atau satu byte. Citra Grayscale dengan range nilai yang lain juga digunakan. Akan tetapi pada umumnya citra tersebut memiliki range pangkat dua dari 8 bit.



Gambar 2.5. Citra Skala keabuan (Gray Scale)

- Citra RGB atau True Color.

Untuk citra RGB masing – masing piksel mempunyai sebuah warna khusus. Warna dideskripsikan oleh kombinasi warna merah (red), hijau (green), dan Biru (blue). Jika masing - masing komponen mempunyai range antara 0-255. Maka total range yang digunakan untuk citra RGB adalah $155 \text{ pangkat } 3$ atau 16.777.216 kemungkinan warna. Range tersebut merupakan range yang mencukupi untuk membuat sebuah citra yang penuh dengan warna. Oleh Karena jumlah bit yang di butuhkan oleh masing – masing piksel adalah 24 maka citra RGB biasa di sebut dengan 24-bit color image. Sebagai sebuah citra yang terdiri dari tumpukan (stuck) 3 matriks yang mewakili nilai merah, hijau, dan biru. Dengan kata lain untuk setiap piksel memiliki 3 buah nilai.



Gambar 2.6. Citra Warna (True Color)

Sedangkan pada citra digital terdapat beberapa elemen dasar pembentuk citra digital, diantaranya :

1. Kecerahan (brightness)

Kecerahan intensitas cahaya rata – rata dari suatu area yang melingkupinya.

2. Kontras (contrast)

Sebaran terang (lightness) dan sebaran gelap (darkness) di dalam suatu citra. Citra dengan kontras rendah komposisi citranya sebagian besar terang atau sebagian besar gelap. Citra dengan kontras yang baik , komposisi gelap dan terangnya tersebar merata.

3. Kontur (Contour)

Merupakan keadaan yang ditimbulkan oleh perubahan intensitas pada piksel – piksel tetangga, sehingga kita dapat mendeteksi tepi objek di dalam citra.

4. Warna (Color)

Adalah persepsi yang dirasakan sistem visual manusia terhadap panjang gelombang cahaya yang di pantulkan objek. Warna – warna yang dapat

ditangkap oleh mata manusia merupakan kombinasi cahaya dengan panjang berbeda. Kombinasi yang memberikan rentang warna paling lebar adalah Red (R), Green (G), Blue (B). (Gunadarma, 2006)

5. Bentuk (shape)

Adalah properti intrinsic dari objek 3 (tiga) dimensi, dengan pengertian bahwa bentuk merupakan property intrinsik utama untuk visual manusia. Umumnya citra yang di bentuk oleh manusia merupakan 2D, sedangkan objek yang dilihat adalah 3D.

6. Tekstur (texture)

Walaupun ada definisi secara formal tentang tekstur, secara intuitif tekstur menyatakan ciri dari permukaan objek yang menggambarkan pola visual. Ciri ini berisi informasi tentang komposisi struktur permukaan, seperti misalnya awan, daun, batu bata, dan kain. Selain itu juga menjelaskan hubungan antara permukaan untuk lingkungan sekitarnya (Al-Teyeche, 2003). Selain itu menurut (Kusao, 2009) Tekstur adalah sifat – sifat atau karakteristik yang dimiliki oleh suatu daerah yang cukup besar sehingga secara alami sifat tersebut dapat berulang dalam daerah tersebut. Sehingga tekstur menjadi salah satu fitur yang penting. Ciri tekstur antara lain meliputi kehalusan (smoothness), kekerasan (coarseness), dan keteraturan (regularity).

Penggunaan fitur tekstur telah banyak digunakan secara luas oleh peneliti dalam menyelesaikan masalah pengenalan pola (pattern recognition) dan

computer vision. Secara umum, representasi tekstur dapat diklasifikasikan menjadi dua, yaitu : structural dan statistic (Gonzales, et al, 2002).

Untuk representasi tekstur secara statistic, dilakukan dengan menganalisa distribusi statistik dari intensitas citra termasuk diantaranya adalah teknik multiresolution filtering seperti Glem Filter.

2.2.5 Pengolahan Citra Digital

Meskipun sebuah citra kaya akan informasi, namun seringkali citra yang kita miliki mengalami penurunan mutu (degradasi), misalnya mengandung cacat atau noise, warnanya terlalu kontras, kurang tajam, kabur (blur), dan sebagainya tentu saja citra semacam ini menjadi lebih sulit diinterpretasi. Karena informasi yang disampaikan oleh citra tersebut menjadi berkurang (Munir, 2004).

Agar citra yang mengalami gangguan mudah diinterpretasi, maka citra tersebut perlu di manipulasi menjadi citra lain yang memiliki kualitas lebih baik. Bidang studi yang menyangkut hal ini adalah pengolahan citra (image processing) (Munir, 2004).

Konsep dasar pemrosesan suatu objek pada gambar menggunakan pengolahan citra diambil dari kemampuan indra penglihatan manusia yang selanjutnya di hubungkan dengan kemampuan otak manusia. Dalam sejarahnya pengolahan citra telah di aplikasikan dalam berbagai bentuk, dengan tingkat kesuksesan yang cukup besar. Seperti berbagai cabang ilmu lainnya, pengolahan citra menyangkut pula berbagai cabang ilmu lainnya, pengolahan citra menyangkut

pula berbagai gabungan cabang – cabang ilmu. Diantaranya adalah optic, electronic, matematika, fotografi, dan teknologi computer (Gonzalez, 2002).

Image processing (pengolahan citra) mencakup dua aspek proses pengubah sebuah citra seperti sebagai berikut :

1. Meningkatkan kualitas informasi dari sebuah citra (gambar) yang digunakan untuk kepentingan interpretasi manusia.
2. Mengubah citra dari sebuah gambar yang digunakan untuk mempermudah pemrosesan persepsi mesin autonomous agar lebih mudah dalam mengambil keputusan.

Pada umumnya, objektivitas dari pengolahan citra adalah mentransformasi atau menganalisa suatu gambar sehingga informasi baru tentang gambar dibuat lebih jelas, Citra dalam pengolahannya dibagi menjadi tiga level proses, yaitu:

1. Low_level process : proses-proses yang berhubungan dengan operasi primitif seperti image pre-processing untuk mengurangi noise, input dan output-nya berupa gambar.
2. Mid-level process : proses-proses yang berhubungan dengan tugas-tugas seperti segmentasi gambar (membagi gambar menjadi objek-objek), pengenalan (recoqnition) suatu objek individu. Pada mid-level process, input pada umumnya berupa gambar tetapi output-nya berupa atribut yang dihasilkan dari proses yang dilakukan gambar tersebut seperti garis, garis contour, dan objek-objek individu.

3. High-level process : proses-proses yang berhubungan dengan hasil dari mid-level proses (Gonzalez, 2002)

Umumnya tahap-tahapan dalam pemrosesan citra untuk mendapatkan informasi, yaitu :

1. Menentukan citra. Pertama kali harus menentukan objek citra yang akan diolah. Untuk kepentingan ini dapat menggunakan kamera atau scanner.
2. Pemrosesan awal. Pemrosesan awal dijalankan sebelum proses utama dijalankan, bagian penting pada tahap ini adalah untuk menjalankan beberapa operasi dasar untuk menghasilkan gambar akhir sehingga lebih mudah dikerjakan pada tahap selanjutnya, Pada kasus ini contoh operasi yang biasa dilakukan adalah mengkonversi citra ke skala keabuan (gray scaling), meningkatkan kontras, dan lainnya.
3. Representasikan dan deskripsi. Pada tahap ini merujuk pada fitur khusus yang memungkinkan pengguna untuk membedakan objek.
4. Pengenalan, klasifikasi, dan interpretasi, pada tahap ini telah didapatkan kode pos dan merepresentasikannya pada sebuah string.

2.2.5.1 Akuisisi citra

Tujuan akuisisi citra adalah menentukan data yang diperlukan dan memilih metode perekaman digital, Tahap ini dimulai dari objek yang akan diambil gambarnya, persiapan alat-alat sampai pada pencitraan. Pencitraan adalah kegiatan transformasi dari citra tempak menjadi citra digital. Beberapa alat yang dapat digunakan seperti : kamera digital, scanner, foto sinar-x atau sinar

infra merah, dan alat penangkapan bayang lainnya. Hasil dari akuisisi citra ini ditentukan oleh kemampuan sensor atau alat untuk mendigitalisasikan sinyal yang terkumpul pada sensor atau alat tersebut,

2.2.5.2 Pra pemrosesan.

Teknik pra-pemrosesan digunakan untuk mempersiapkan citra agar dapat menghasilkan ciri yang lebih baik pada tahap pemisahan ciri (ekstraksi ciri) terhadap proses pengenalan pola. Pra-pemrosesan adalah pengubahan data mentah untuk membantu kemampuan komputasional dan pencari ciri. Pada tahap pra-pemrosesan, kita cukup mengkaji metode atau pendekatan fungsi yang tepat untuk melakukan pemrosesan awal (pra-pemrosesan).

2.2.5.2.1 Konversi Citra Derajat Keabuan (gray scalling)

Gray scalling adalah teknik yang digunakan untuk mengubah citra berwarna menjadi bentuk tingkat aras keabuan. Jumlah warna citra grey adalah 256, karena citra grey jumlah bit-nya adalah 8, sehingga jumlah warnanya adalah $2^8=256$, nilainya berada pada jangkauan 0-255. Pengubahan citra warna ke citra skala keabuan dapat menggunakan beberapa algoritma diantaranya : averaging, Luminosity, Desaturation, citra minimal dan Maximal decomposition, single color channel.

Metode Luminosity ini menggunakan perhitungan nilai berdasarkan tingkat luminositas. Metode ini juga bekerja dengan cara merata-ratakan nilai, tetapi metode ini membentuk rata-rata dengan bobot tertentu berdasarkan persepsi mata manusia. Mata manusia lebih sensitif terhadap warna hijau, oleh karena

itu warna hijau diberi bobot yang lebih besar dari pada warna lainnya, Metode luminosity yang bekerja menggunakan persepsi mata manusia menghasilkan citra yang paling baik dari beberapa metode yang digunakan untuk konversi citra ke skala keabuan.

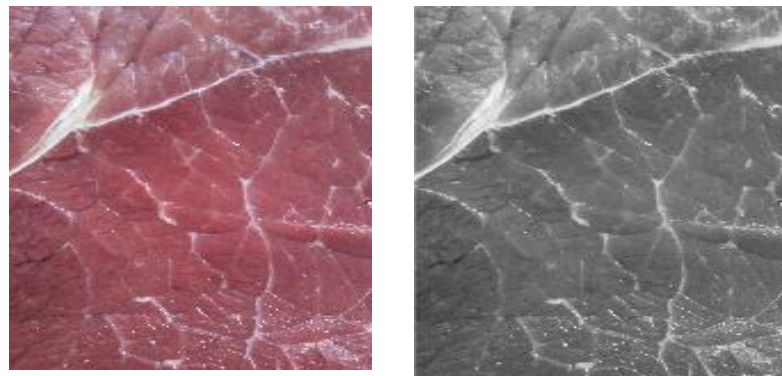
Adapun beberapa persamaan dari metode luminosity sebagai berikut :

$$P1 = (0,21R)+(0.71G)+(0.07B)$$

$$P2=(0.2125R)+(0.7154G)+(0.0721B)$$

$$P3=(0.50R)+(0.419G)+(0.081B)$$

$$P4=(0.229R)+(0.587G)+(0.114B)$$



Gambar 2.7 Hasil mengubah citra RGB ke citra grayscale. (a)

Citra RGB/Asli, dan (b) Citra Grayscale/hasil pengubahan

2.2.5.2.2 Ekualisasi histogram (histogram aqualization)

Metode ekualisasi histogram merupakan teknik penyesuaian nilai piksel sehingga menghasilkan citra dengan kontras yang lebih baik. Metode ini menggunakan perataan histogram agar derajat keabuan dari yang paling rendah (nol) hingga ke yang paling tinggi (255) memiliki kemunculan yang rata. Dengan ekualisasi histogram, hasil gambar yang memiliki histogram

yang tidak merata atau distributif kualitatif yang memiliki banyak loncatan gradasinya akan menjadi gambar yang lebih jelas karena derajat keabuan tidak domain gelap atau domain terang.

2.2.5.3 Ekstraksi fitur (feature extraction).

Ekstraksi fitur (ciri) adalah proses untuk mendapatkan ciri-ciri pembela yang membedakan suatu sampel dari sampel lainnya. Bagi sebagian besar aplikasi pengenalan pola (pattern recognition), teknik ekstraksi fitur yang handal merupakan kunci utama dalam penyelesaian masalah pengenalan pola. Dengan adanya ekstraksi fitur, informasi penting yang berada dalam citra tersebut dapat diambil dan disimpan kedalam vektor fitur (feature vektor).

Fitur-fitur yang dapat diekstrak pada citra dapat berdasarkan elemen warna, bentuk, tekstur, kecerahan, kontur, dan kontras dengan menggunakan teknik ekstraksi fitur tertentu.

2.2.5.3.1 Ekstraksi fitur berdasarkan tekstur

Tekstur merupakan karakteristik instrinsik dari suatu citra yang terkait dengan tingkat kekerasan (roughness), granularitas (granulation), dan keteraturan (regularity) susunan struktur piksel : Tekstur dicirikan sebagai distribusi spasial dari derajat keabuan didalam sekumpulan piksel-piksel yang bertetangga. Jadi tekstur, tidak dapat diindeksasikan untuk sebuah piksel, melainkan suatu citra dianggap sebagai suatu kesatuan.

Informasi tekstur dapat digunakan untuk membedakan sifat permukaan tersebut, sehingga analisis tekstur lazim dimanfaatkan sebagai proses antara untuk melakukan klasifikasi dan interpretasi citra.

Ekstraksi fitur citra berdasarkan tekstur pada orde pertama dapat menggunakan mode statik, yaitu dengan melihat statistik distribusi derajat keabuan pada histogram citra tersebut (Wong & Zrimec, 2006). Dalam penelitian ini nilai-nilai pada histogram yang dihasilkan dihitung dari parameter ciri (fitur) Entropy. Entropy (H) menunjukkan ukuran tingkat keacakan (ketidakteraturan) bentuk dari suatu citra. Entropy dirumuskan sebagai berikut:

$$H = - \sum_{n=0}^n p(fn)^2 \log p(fn)$$

Dimana :

fn =Nilai intensitas keabuan

$p(fn)$ = Nilai histogram

2.2.5.3.2 Ekstraksi Fitur Berdasarkan Warna

Untuk membedakan suatu objek dengan warna tertentu dapat menggunakan nilai hue yang merupakan representasi dari cahaya tampak (Merah,jingga,kuning,hijau,biru,ungu). Nilai hue dapat dikombinasikan dengan nilai saturation dan value yang merupakan tingkat kecerahan suatu warna. Untuk mendapatkan ketiga nilai tersebut, perlu dilakukan konversi

ruang warn citra yang semula RGB (Red, Green, Blue) menjadi HSV (Hue, Saturation, value) melalui persamaan berikut.

$$R' = R/255$$

$$G' = G/255$$

$$B' = B/255$$

$$C_{\max} = \max (R', G', B')$$

$$C_{\min} = \min (R', G', B')$$

$$\Delta = C_{\max} - C_{\min}$$

2.2.5.3.3 Gray level Co-occurrence Matrix

Fitur Ekstraksi Grey Level Co-occurrence Matrix (GLCM) merupakan matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak d dan orientasi arah sudut θ tertentu dalam citra, jarak dinyatakan dalam piksel dan orientasi sudut dinyatakan dalam derajat

Hubungan
pada matriks
matriks:

0	0	1	1
0	0	1	1
0	2	2	2
2	2	2	2

2.1.contoh matrix asal
untuk $\theta=0^\circ$ dan $d=1$
dapat dituliskan dalam

2	2	1	0
0	2	0	0
0	0	3	1
0	0	0	1

Tabel 2.2. Matrix Co-Occurrence dari Matrix Asal

Matriks Co-occurrence yang dapat kemudian di tambahkan dengan matriks transposenya untuk menjadikanya simetris terhadap sumbu diagonal

0.250	0	0.083	0
0	0.167	0.083	0
0.083	0.083	0.083	0.083
0	0	0.083	0

Tabel 2.3. GLCM

simetris ternormalisasi

dari matriks asal

Langkah – langkah membuat GLCM simetris ternormalisasi adalah sebagai berikut :

- Membuat Framework Matriks
- Menentukan hubungan spasial antara piksel referensi dengan piksel tetangga, berupa sudut dan jarak.
- Menghitung jumlah co-occurrence dan mengisikanya pada framework.
- Menjumlahkan matriks Co-occurrence transposnnya untuk menjadikannya simetris.
- Normalisasi matriks untuk mengubahnya ke bentuk probabilitas

Dalam penelitian ini Ciri – ciri fitur di ekstraksi GLCM yang di gunakan berjumlah enam, yaitu:

➤ Angular second moment

Menunjukkan sifat kehomogenitas citra.

$$ASM = \sum_i \sum_j \{p(i,j)\}^2$$

➤ Contrast

Menunjukkan ukuran penyebaran elemen – elemen matriks citra.

$$CON = \sum_k k^2 \left[\sum_i \sum_j (i,j) \right] |i - j| - k$$

➤ Corelation

Menunjukkan ukuran ketergantungan linear derajat keabuan citra sehingga dapat memberikan petunjuk adanya struktur linear dalam citra.

$$COR = \frac{\sum_i \sum_j j(i,j).p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$

➤ Variance

Menunjukkan variasi elemen – elemen matriks kookurensi. Citra dengan transisi derajat keabuan kecil kan memiliki variasi yang kecil pula.

$$VAR = \sum_i \sum_j (i - \mu_x)(j - \mu_y)P(i,j)$$

➤ Inverse Difference Moment

Menunjukkan kehomogenan citra yang berderajat keabuan sejenis citra homogeny memiliki harga IDM yang besar.

$$IDM = \sum_i \sum_j \frac{1}{1 + (i - j)^2} P(i, j)$$

➤ Entrophy

Menunjukkan ukuran ketidak teraturan bentuk, harga ENT besar untuk citra dengan transisi derajat keabuan merata dan bernilai kecil jika struktur citra dapat diatur.

$$ENT_2 = - \sum_i \sum_j p(i, j) \cdot \log p(i, j)$$

Pada tahap ekstraksi ciri GLCM menggunakan empat fitur yaitu Correlation, contrast, homogeneity dan energy serta GLCM memiliki variasi empat sudut yaitu 0°, 45°, 90°, 135°. Kemudian hasil dari matriks GLCM tersebut yang akan digunakan dalam klasifikasi pada tahap klasifikasi menggunakan KNN

2.2.5.4 Klasifikasi (classification)

Klasifikasi adalah penemuan sebuah model (atau fungsi) yang menggambarkan dan membedakan kelas data atau konsep yang bertujuan agar bias digunakan untuk memprediksi kelas dari objek yang label kelasnya (Han, 2006). Sedangkan klasifikasi menurut serapannya yang berasal dari belanda, classificatie, yang sendirinya berasal dari Bahasa perancis classification.

Istilah ini menunjuk kepada sebuah metode untuk menyusun data secara sistematis atau menurut beberapa aturan atau kaidah yang telah ditetapkan.

Algoritma klasifikasi yang banyak digunakan, yaitu Decision/Classification Trees, Bayesian Classifiers/ Naïve Bayesian Classifiers, Neural Network, Analisa statistic, algoritma genetika, rough sets, K-Nearest Neighbor, metode rule base, Memory Based Reasoning, dan support vector machines (SVM).

Klasifikasi data terdiri dari dua tahap proses. Pertama adalah Learning (fase training), dimana algoritma klasifikasi dibuat untuk menganalisa data training lalu diintreprentasikan dalam bentuk rule klasifikasi. Proses kedua adalah klasifikasi, dimana data tes digunakan untuk memperkirakan akurasi dari rule klasifikasi (Han, 2006).

Model dalam klasifikasi sama artinya dengan suatu model yang menerima masukan kemudian mampu melakukan pemikiran terhadap masukan dan memberikan jawaban sebagai keluaran dari hasil pemikiran terhadap masuk dan memberikan jawaban sebagai keluaran hasil pemikiran, Proses klasifikasi didasarkan pada empat komponen (Gurunescu, 2011) :

1. Kelas adalah variable dependen yang berupa kategorikal yang mempresentasikan “label” yang terdapat pada objek.
2. Predictor, merupakan variable independen yang direpresentasikan oleh karakteristik (attribute) data.
3. Training dataset, Satu set data yang berisi nilai dari dua komponen diatas yang digunakan untuk menentukan kelas yang cocok berdasarkan predictor.

4. Testing Dataset, Berisi data baru yang akan di klasifikasikan oleh model yang telah di buat dan akurasi klasifikasi dievaluasi.

2.2.5.4.3 k-Nearest Neighbors (k-NN)

Algoritma k-NN adalah suatu metode yang menggunakan algoritma *supervised*.

Tujuan dari algoritma k-NN adalah untuk mengklasifikasi objek baru berdasarkan atribut dan data training.

Algoritma k-NN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari sampel uji yang baru, misalnya pada penelitian Algoritma k-Nearest Neighbor dalam Klasifikasi Data Hasil Produksi Kelapa Sawit pada PT. Minamas .

(Krisandi, Helmi, & Prihandono, 2013)

Jarak yang digunakan adalah jarak *Cosine Similarity Measure*, yang didefinisikan sebagai berikut .

$$\text{Similarity}(d_j, q) = \frac{d_j \cdot q}{|d_j| \cdot |q|} = \frac{\sum_{i=1}^t w_{ij} \cdot w_{iq}}{\sqrt{\sum_{i=1}^t w_{ij}^2 \sum_{i=1}^t w_{iq}^2}}$$

Keterangan:

d_j = ciri data *learning*

q = ciri data *query*

w_{ij} = vektor ciri ke-i data *learning*

w_{iq} = vektor ciri ke-i data *query*

Contoh Kasus Perhitungan K-NN

Terdapat beberapa data yang berasal dari survey questioner tentang klasifikasi kualitas kertas tissue apakah baik atau jelek, dengan objek training dibawah ini menggunakan dua attribute yaitu daya tahan terhadap asam dan kekuatan.

X1=Daya tahan Asam(detik)	X2=Kekuatan (Kg/m2)	Y=Klasifikasi
8	4	Baik
4	5	Jelek
4	6	Jelek
7	7	Baik
5	6	Jelek
6	5	Baik

Tabel 2.4 Contoh Kasus

Akan diproduksi kembali kertas tissue dengan attribute **X1=7** dan **X2=4**, tanpa harus mengeluarkan biaya untuk melakukan survey, maka dapat diklasifikasikan kertas tissue tersebut termasuk yang baik atau jelek.

X1=Daya tahan Asam(detik)	X2=Kekuatan (Kg/m2)	Square Distance to query distance (7,4)
8	4	$(8-7)^2+(4-4)^2=1$
4	5	$(4-7)^2+(4-4)^2=10$
4	6	$(4-7)^2+(6-4)^2=13$
7	7	$(7-7)^2+(7-4)^2=9$
5	6	$(5-7)^2+(6-4)^2=8$
6	5	$(6-7)^2+(5-4)^2=2$

Tabel 2.5 Contoh Hasil Klasifikasi

X1=Daya tahan Asam(detik)	X2=Kekuatan (Kg/m2)	Square Distance to query distance (7,4)	Jarak Terkecil	Apakah termasuk nearest neighbor(K)	Y=Kategori nearest neighbor
8	4	$(8-7)^2+(4-4)^2=1$	1	Ya	Baik
4	5	$(4-7)^2+(4-4)^2=10$	5	Tidak	-
4	6	$(4-7)^2+(6-4)^2=13$	6	Tidak	-

7	7	$(7-7)^2+(7-4)^2=9$	4	Ya	Baik
5	6	$(5-7)^2+(6-4)^2=8$	3	Ya	Jelek
6	5	$(6-7)^2+(5-4)^2=2$	2	Ya	Baik

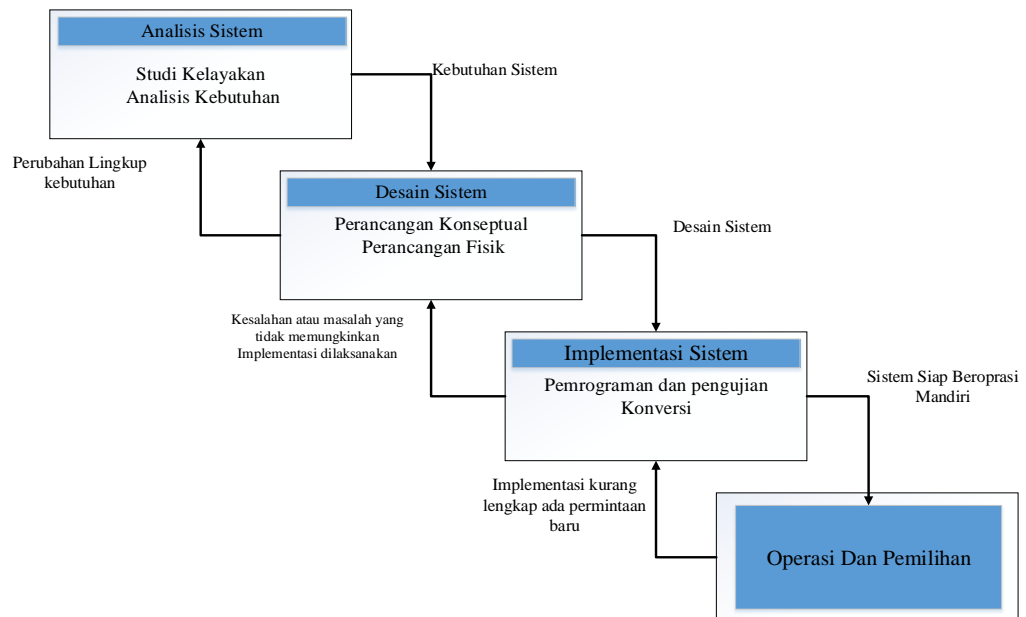
Tabel 2.6 Hasil AKhir

Dengan mengurutkan jarak terkecil, semisal diambil $K=3$, maka perbandingan nya adalah 2 (Baik) > 1 (Jelek). Maka dapat disimpulkan kertas tissue dengan attribute **X1=7** dan **X2=4** masuk ke kelas **Baik**.

2.3 Siklus Pengembangan Sistem

Bila dalam operasi sitem yang sudah dikembangkan masih timbul permasalahan-permasalah yang tidak dapat diatasi dalam tahap pemeliharaan sistem, maka perlu dikembangkan kembali ke siatu sistem untuk mengatasinya dan proses ini kembali ke proses yang pertama. Siklus ini disebut dengan siklus hidup suatu sistem. Siklus hidup pengembangan sistem dapat di idefinisikan sebagai serangkaian aktivitas yang di laksanakan oleh professional dalam memakai sistem informasi untuk mengembangkan daam mengimplementasikan sistem informasi.

Segala sesuatu yang akan kita kembangkan seharusnya memiliki kerangka kerja, demikian pula dengan langkah-langkah pengembangan sistem/perangkat lunak (Adi Nugroho. 2010;3)



Gambar 2.8 Siklus Hidup Pengembangan Sistem

2.2.6. Analisis

Model analisis perangkat lunak, seperti kita ketahui, sesungguhnya merupakan spesifikasi rinci dari kebutuhan – kebutuhan pengguna dan bekerja sebagai langkah pertama saat kita kelak akan mengembangkan model perancangan perangkat lunak (Nugroho, 2010 : 87).

Setelah proses perencanaan atau planning tahapan selanjutnya adalah analisis terhadap data – data mengenai kebutuhan pengguna yang di padukan dengan hasil studi kelayakan yang meliputi data – data ketersediaan sumber daya yang ada seperti manusia, infrastuktur teknologi yang tersedia dan lain -lain, yang kemudian menetapkan solusi yang tepat untuk diterapkan. Pada tahapan ini seorang analis sistem melakukan pengenalan terhadap segenap permasalahan yang timbul pada pengguna melalui dekoposisi dan realisasi permasalahan lebih lanjut, mengenai komponen – komponen sistem/perangkat lunak, objek – objek,

hubungan antar objek, dan sebagainya, dimana secara keseluruhan dinotasikan dalam bentuk use case.

2.2.7. Perancangan

Menurut Rober J. Verzello/John Reuter III Definisi Sistem:

“Tahap setelah analisis dari siklus pengembangan sistem :

Pendefinisian dari kebutuhan – kebutuhan fungsional dan persiapan untuk merancang bangun implementasi; menggambarkan bagaimana suatu sistem dibentuk”.

Hariyanto (2004 : 421) mengungkapkan, perancangan (design) sistem adalah strategi tingkat tinggi untuk membuat keputusan – keputusan berikut:

- a) Pengorganisasian sistem menjadi subsistem – subsistem.
- b) Alokasi subsistem ke komponen – komponen perangkat keras dan perangkat lunak.
- c) Keputusan – keputusan konseptual dan kebijakan utama untuk membentuk kerangka kerja rancangan inti.

a. Analisis sitem berorientasi objek

Hasil dari analisis berorientasi objek merupakan suatu masukan bagi perancangan sistem/perangkat lunak berorientasi objek. Pada model analisis ini memungkinkan ketersedianya informasi detail mengenai skenario penggunaan sistem dan menetapkan landasan untuk model implementasi dan lingkungan dengan mengidentifikasi elemen struktur sistem. Kelas – kelas objek yang meliputi atribut – atribut, metode – metode, dan asosiasi yang

didapatkan dari hasil analisis merupakan suatu kerangka kerja saat perancangan dan analisis tersebut harus dirancang untuk implementasi.

b. Perancangan system

Perancangan sistem merepresentasikan arsitektur perangkat lunak secara umum yakni:

1. Arsitektur konseptual berkaitan dengan model kelas statik dan koneksi – koneksi di antara komponen – komponen model.
2. Arsitektur model mendeskripsikan pembagian sistem menjadi subsistem – subsistem atau modul – modul ini saling berkomunikasi.
3. Arsitektur kode mendefinisikan bagaimana kode program diorganisasikan menjadi berkas – berkas dan direktori – direktori dan dikelompokkan menjadi pustaka – pustaka.
4. Arsitektur eksekusi berfokus pada aspek dinamis dari sistem dan komunikasi antara komponen – komponen sebagai task dan operasi yang dieksekusi.

c. Perancangan Objek

Perancangan objek berfokus pada deskripsi objek dan interaksinya antara satu objek dengan objek yang lain secara rinci. Untuk spesifikasi rinci dari struktur data untuk atribut dan rancangan prosedural dari semua operasi diciptakan selama perancangan objek. Sementara ketampakan dari semua atribut kelas harus didefinisikan dan antarmuka antara objek-objek dikolaborasikan untuk mendefinisikan rincian-rincian dari model perawatan pesan secara lengkap.

d. Perancang antar muka manusia

Pandangan model pemakai menuntun ke proses perancangan antarmuka pemakai, menyediakan skenario yang dikolaborasikan secara literasi untuk menjadikannya sebagai sekumpulan kelas-kelas antar muka.

e. Perancangan Manajemen data

Perancangan manajemen data adalah untuk mendukung persisten objek yang menetapkan sekumpulan klas-kelas dan kolaborasi-kolaborasi yang memungkinkan sistem mengolah persisten data (berkas-berkas dan basis data) yang mengimplementasikan persisten objek, dimana pada umumnya manajemen data dirancang secara berlapis, Perancangan manajemen data ini meliputi perancang atribut-atribut dan operasi-operasi yang diperlukan untuk mengelola objek-objek.

f. Perancangan Manajemen Task

Perancangan Manajemen Task menetapkan infrastruktur yang mengorganisasikan subsistem-subsistem menjadi task dan kemudian mengelola konkurensi task.

Perancangan (desain) sistem pada pengembangan sistem/perangkat lunak yang berorientasi objek pada umumnya menggunakan bahasa modeling UML (Unified Modeling Language) dan menyajikan dalam bentuk diagram-diagram yang nantinya akan menjadi acuan bagi programmer untuk melakukan coding (pembangunan program). Adapun diagram-diagram tersebut meliputi seperti yang disajikan pada tabel berikut:

Table 2.7. Daftar pengelompokkan diagram pada UML.

No	Nama Diagram	KATEGORI	Deskripsi
1	Class	Struktur	Mengilustrasikan sekumpulan kelas, paket dan hubungan yang merinci satu aspek tertentu dari sistem.
2	Object	Struktur	Snapshot dari sistem yang mengilustrasikan hubungan static yang terjadi diantara objek – objek.
3	Component	Struktur	Berkaitan dengan hubungan statistik yang ada antara komponen – komponen perangkat lunak yang di pasang. Contohnya file exe, dll, ocx, beans, dan sebagainya.
4	Deployment	Struktur	Mendeskripsikan komponen fisik sistem. Umumnya termasuk beragam simpul pengolahan, direalisasikan dalam bentuk perangkat (Printer, Modem) atau pemproses (server).

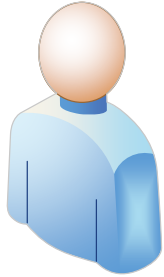
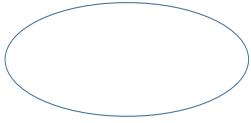
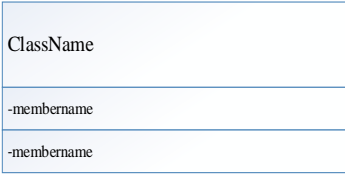
5	Use Case	Perilaku	Menunjukkan sekumpulan data use case dan hubungan antara keduanya. Diagram use case berkontribusi ke organisasi model yang efektif juga merupakan permodelan inti perilaku sistem.
6	Activity	Perilaku	Memodelkan aktifitas di antara proses – proses. Kebanyakan berguna untuk merinci perilaku use case
7	State	Perilaku	Mengilustrasikan perilaku yang berhubungan dengan state dari objek. Transisi – transisi di antara state –state membantu mengidentifikasi dan memvalidasi perilaku kompleks.
8	Sequence	Perilaku	Tipe diagram interaksi yang mendeskripsikan pesan – pesan berbasis waktu yang dikirim objek – objek.
9	Collaboration	Perilaku	Tipe diagram interaksi yang mendeskripsikan layout organisasi

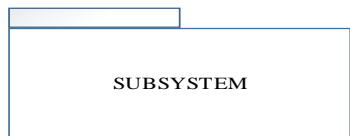
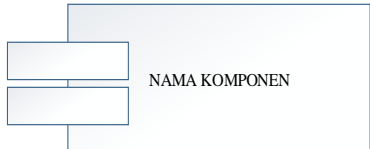

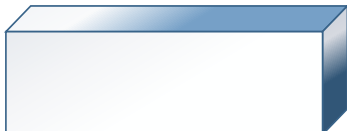
			dari objek – objek yang mengirim dan menerima pesan.
--	--	--	--

Sumber : (Harianto, 2004: 335-336)

Berikut beberapa notasi pengklasifikasi (*classifier*) yang digunakan dalam Bahasa *modelling* UML seperti di sajikan pada table berikut :


Table 2.8. Notasi pengklasifikasian dalam UML

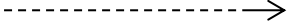

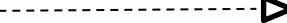
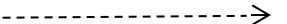
Pengklasifikasian	Kegunaan	Notasi
Actor	Menggambarkan semua objek di luar sistem (bukan hanya pengguna sistem/perangkat lunak) yang berinteraksi dengan sistem yang di kembangkan.	
Use Case	Menggambarkan fungsionalitas yang dimiliki sistem.	
Kelas (Class)	Menggambarkan konsep dasar permodelan sistem.	

Subsistem (subsystem)	Menggambarkan paket spesifikasi serta implementasi	
Komponen (Component)	Menggambarkan bagan – bagan fisik sistem/perangkat lunak yang dikembangkan.	
Antarmuka (Interface)	Menggambarkan antarmuka pengiriman pesan (message) antar pengklasifikasi	
Simpul (Node)	Menggambarkan sumber daya komputasi yang digunakan oleh system	

Sumber : (Nugroho, 2010:16)

Table 2.9. Notasi relasi – relasi dalam UML

Relasi	Fungsi	Notasi
Asosiasi (Association)	Mendesripsikan hubungan antara instance suatu kelas	

Ketergantungan (Dependency)	Relasi antara dua elemen model.	
Generalisasi (Generalization)	Relasi antara dua pengklasifikasi yang memiliki deskripsi yang bersifat lebih umum dengan berbagai pengklasifikasi yang lebih spesifik.	
Realisasi (realization)	Relasi antara spesifikasi dan implementasinya.	
Pengguna (Usage)	Situasi di mana salah satu elemen membutuhkan elemen yang lain agar dapat berfungsi dengan baik	

Sumber : (Nugroho, 2010:2

2.2.8. Implementasi

Tahapan implementasi merupakan tahap dimana dilakukan transformasi/penerjemahan dari Bahasa modelling ke suatu Bahasa pemrograman hal ini merupakan tugas dari programmer, pada pengembangan sistem/perangkat lunak berorientasi objek penerjemah dari setiap diagram – diagram UML yang

telah dirancang pada tahap perancangan harus diterjemahkan ke dalam Bahasa pemrograman sama persis dengan diagram – diagram yang ada guna menghindari terjadinya perubahan fungsi/tujuan dari pengembangan sistem/perangkat lunak.

2.2.9. Pengujian

Pada pendekatan berorientasi objek, pengujian merupakan sesuatu persoalan yang lebih kompleks dibandingkan dengan pendekatan konvensional. Karena keberadaan pewarisan, polymorphism, dan pengkapsulan pada pengembangan sistem berorientasi objek menimbulkan suatu persoalan yang baru untuk perancangan kasus pengujian dan analisis hasil.

Hariyanto (2004 : 583) mengungkapkan bahwa : fitur – fitur berikut berpengaruh dalam teknik – teknik pengujian yang perlu dilakukan.

- a) Pengkapsulan (encapsulation)
- b) Penyusunan objek – objek (Object Composition)
- c) Pewarisan (inheritance)
- d) Interaksi (Interaction)
- e) Polymorphism
- f) Pengikat Dinamis (Dynamic Binding)
- g) Guna ulang (Reuse)
- h) Genericity dan kelas abstrak

Dari kompleksnya fitur – fitur yang mempengaruhi dalam pengujian sistem berorientasi objek maka strategi pengujian dilakukan pada :

- a) Pengujian Unit, dimana pengujian unit dilakukan hingga beberapa level dengan alasan adanya konsep pewarisan. Pengujian unit ini bertujuan

- b) untuk menjamin setiap unit memenuhi spesifikasi. Kelas – kelas merupakan sasaran pengujian unit.
- c) Pengujian integrasi, pengujian ini dilakukan untuk memverifikasi implementasi dari satu use case yang telah bekerja seperti yang telah di harapkan.
- d) Pengujian validasi, pengujian ini dilakukan untuk menjamin fungsi – fungsi sistem/aplikasi telah dilakukan secara benar, pengujian dieksekusi ketika satu sistem (sub sistem) yang lengkap telah di rakit. Pengujian validasi ini meliputi rincian – rincian objek yang tidak tampak, focus pada masukan dan keluaran yang tampak oleh pemakai.

2.2.10. Pemeliharaan

Pemeliharaan adalah proses pengubahan sistem setelah beroperasi dan digunakan (Hariyanto 2004 : 603).

Pemeliharaan sistem adalah tahap dimana kita mulai pengoprasian sistem dan jika diperlukan melakukan perbaikan – perbaikan kecil (Nugroho, 2010 : 7).

Dari ke dua pendapat di atas memiliki pandangan yang sama dimana setelah sistem di oprasikan dapat saja dilakukan perubahan, apakah itu sebagian kecil dari sistem ataupun secara keseluruhan dengan harapan untuk dapat memenuhi kebutuhan dari pengguna sistem tersebut.

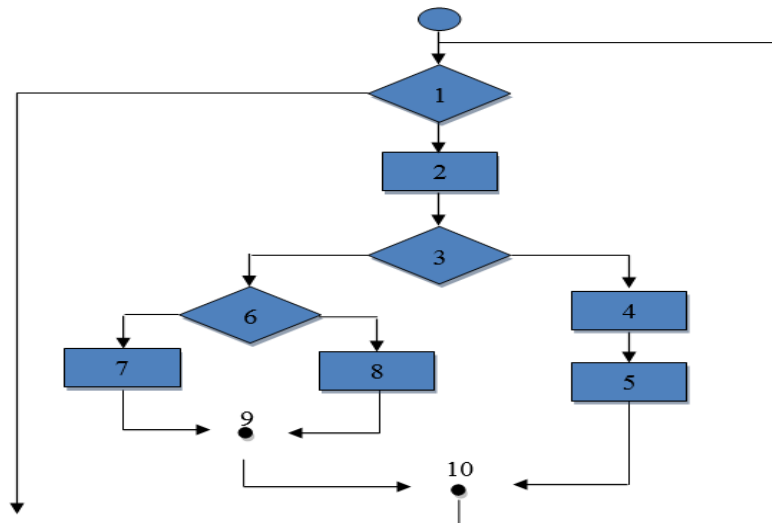
2.3. Teknik Pengujian Sistem

Pengujian perangkat lunak merupakan pengujian terhadap logika program. Adapun metode yang dapat digunakan untuk menguji perangkat lunak adalah

metode white box testing dan black box testing. Pengujian perangkat lunak merupakan elemen kritis dari jaminan kualitas suatu perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi desain dan pengkodean (Pressman, 2002).

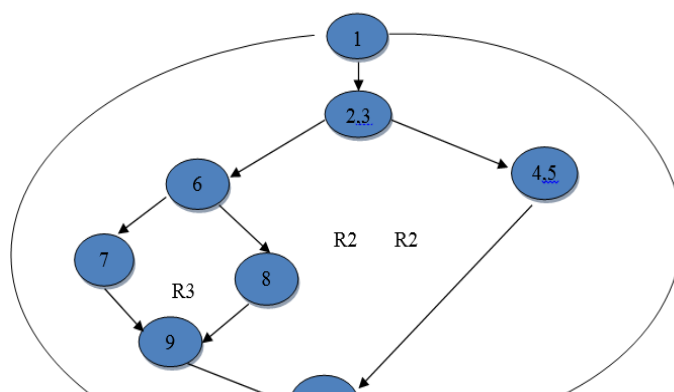
2.3.1. White box

Pengujian white box adalah metode pengujian yang menggunakan struktur kontrol desain prosedur untuk memperoleh test case. Dengan menggunakan metode white box, perekayasa sistem dapat melakukan test case yang memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali, menggunakan semua keputusan logis pada sisi true dan false, mengeksekusi semua loop pada batasan mereka dan pada batas operasional mereka, dan menggunakan struktur data internal untuk menjamin validitasnya. Pengujian basis path adalah teknik pengujian white box yang diusulkan pertama kali oleh Tom McCabe. Metode basis path ini memungkinkan desain test case mengukur kompleksitas logis dari desain prosedural dan menggunakannya sebagai pedoman untuk menentukan basis set dari jalur eksekusi (Roger S. Pressman, 2002).



Gambar 2.9. Bagan Alir

Bagan alir digunakan untuk menggambarkan struktur kontrol program dan untuk menggambarkan grafik alir, harus memperhatikan representasi desain prosedural pada bagan alir. Pada gambar dibawah ini, grafik alir memetakan bagan alir tersebut ke dalam grafik alir yang sesuai (dengan mengasumsikan bahwa tidak ada kondisi senyawa yang diisikan di dalam diamond keputusan dari bagan alir tersebut). Masing-masing lingkaran, yang disebut *simpul* grafik alir, merepresentasikan satu atau lebih statemen prosedural. Urutan kotak proses dan permata keputusan dapat memetakan simpul tunggal. Anak panah tersebut yang disebut *edges* atau *links*, merepresentasikan aliran kontrol dan analog dengan anak panah bagan alir. *Edge* harus berhenti pada suatu simpul, meskipun bila simpul tersebut tidak merepresentasikan statemen prosedural. (Roger S. Pressman, 2002. 536)



Node adalah lingkaran yang merepresentasikan satu atau lebih statemen prosedural.

Edge adalah anak panah pada grafik alir

Region adalah area yang membatasi *edge* dan *node*

Simpul Predikat adalah simpul *node* yang berisi kondisi yang ditandai dengan dua atau lebih *edge* yang berasal darinya.

Dari gambar *flowgraph* di atas di dapat :

Jalur 1 : 1 – 11

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 10 – 1 – 11

Jalur 3 : 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Jalur 4 : 1 – 2 – 3 – 6 – 7 – 9 – 10 – 1 – 11

Jalur 1, 2, 3, dan 4 yang ditentukan di atas terdiri dari sebuah *basis set* untuk grafik alir pada gambar 2.9 Bagaimana kita tahu banyaknya jalur yang dicari? Komputasi kompleksitas siklomatis memberikan jawaban. Fondasi kompleksitassiklomatis adalah teori grafik, dan memberi kita metriks perangkat lunak yang sangat berguna. Kompleksitas dihitung dalam salah satu dari tiga cara berikut :

1. Jumlah region grafik alir sesuai dengan kompleksitas siklomatis.
2. Kompleksitas siklomatis, $V(G)$, untuk grafik alir G ditentukan sebagai

$V(G) = E - N + 2$ di mana E adalah jumlah *edge* grafik alir dan N adalah jumlah simpul grafik alir.

3. Kompleksitas siklomatis, $V(G)$, untuk grafik alir G juga ditentukan sebagai $V(G) = P + 1$, dimana P adalah jumlah simpul predikat yang diisikan dalam grafik alir G .

Pada gambar 2.8 grafik alir, kompleksitas siklomatis dapat dihitung dengan menggunakan masing-masing dari algoritma yang ditulis di atas :

1. Grafik alir mempunyai 4 region.
2. $V(G) = 11 \text{ edge} - 9 \text{ simpul} + 2 = 4$.
3. $V(G) = 3 \text{ simpul yang diperkirakan} + 1 = 4$.

Dengan demikian, kompleksitas siklomatis dari grafik alir pada gambar 2.9 adalah

4. Yang lebih penting, nilai untuk $V(G)$ memberi kita batas atas untuk jumlah jalur independen yang membentuk *basis set*, dan implikasinya batas atas.

2.3.2 Black Box

Black box approach adalah suatu sistem dimana *input dan output*-nya dapat didefinisikan tetapi prosesnya tidak diketahui atau tidak terdefinisi. Metode ini hanya dapat dimengerti oleh pihak dalam (yang menangani sedangkan pihak luar hanya mengetahui masukan dan hasilnya). Sistem ini terdapat pada subsistem tingkat terendah.

Metode uji coba *black box* memfokuskan pada keperluan fungsional dari *software*. Karena itu ujicoba *black box* memungkinkan pengembang *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program. Uji coba *black box* bukan merupakan alternatif dari uji coba *white box*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *white box*.

Uji coba *black box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database eksternal*
4. Kesalahan performa
5. Kesalahan inisialisasi dan terminasi

Tidak seperti metode *white box* yang dilaksanakan diawal proses, uji coba *black box* diaplikasikan di beberapa tahapan berikutnya. Karena uji coba *black box* dengan sengaja mengabaikan struktur kontrol, sehingga perhatiannya difokuskan pada informasi *domain*. Uji coba didesain untuk dapat menjawab pertanyaan-pertanyaan berikut :

1. Bagaimana validitas fungsionalnya diuji ?
2. Jenis *input* seperti apa yang akan menghasilkan kasus uji yang baik ?
3. Apakah sistem secara khusus sensitif terhadap nilai input tertentu ?
4. Bagaimana batasan – batasan kelas data isolasi ?
5. Berapa rasio data dan jumlah data yang dapat ditoleransi oleh sistem ?
6. Apa akibat yang akan timbul dari kombinasi spesifik data pada operasi sistem ?

Dengan mengaplikasikan uji coba *black box*, di harapkan dapat menghasilkan sekumpulan kasus uji yang memenuhi kriteria berikut :

1. Kasus uji yang berkurang, jika jumlahnya lebih dari 1, maka jumlah dari uji kasus tambahan harus didesain untuk mencapai uji coba yang cukup beralasan.
2. Kasus uji yang memberitahukan sesuatu tentang keberadaan atau tidaknya suatu jenis kesalahan, dari pada kesalahan yang terhubung hanya dengan suatu uji coba yang spesifik.

2.4 Pengujian Model

Sebuah sistem yang melakukan klasifikasi diharapkan dapat melakukan klasifikasi semua dataset semua data set dengan benar, tetapi tidak dapat di pungkiri bahwa kinerja suatu sistem tidak biasa 100% benar, sehingga sebuah sistem klasifikasi harus diukur kinerjanya (Prasetyo, 2012). Umumnya, pengukuran kinerja klasifikasi dilakukan dengan Confusion matrix (matrix konfusi) atau root mean Squared Error (RMSE).

Pada penelitian ini peneliti menggunakan confusion matrix. Pengujian confusion matrix merepresentasikan hasil evaluasi mode dengan menggunakan table matriks, jika data set terdiri atas dua kelas, kelas pertama dianggap positif, dan kelas kedua dianggap negative. Evaluasi menggunakan confusion matrix menghasilkan nilai akurasi, presisi, recall, akurasi dalam klasifikasi merupakan persentase ketepatan record data yang diklasifikasikan secara benar setelah dilakukan pengujian pada hasil kasifikasi. Precision atau confidence merupakan proporsi kasus yang diprediksi positif yang juga positif benar pada data yang sebenarnya. Recall atau sensitivity merupakan propose kasus positif yang sebenrnya diprediksi secara benar

Tabel 2.10. Confusion Matrix

Correct Clasification	Classified as	
	+	-
+	True Positives	False Positives
-	False Positives	True Positives

True positif (tp) merupakan jumlah record positif dalam dataset yang diklasifikasikan positive. True negatives (tn) merupakan jumlah record negative dalam dataset yang diklasifikasikan negative. False positive (fp) merupakan jumlah record positif dalam dataset yang diklasifikasikan negative.

Berikut persamaan model confusion matrix:

1. Nilai akurasi (acc) adalah proporsi jumlah prediksi yang benar, dapat dihitung dengan persamaan :

$$acc = \frac{tp + tn}{tp + tn + fp + fn}$$

2. Sensitivity atau recall digunakan untuk membanding proporsi tp terhadap tupel yang positif, yang dihitung dengan persamaan:

$$recall = \frac{tp}{tp + fn}$$

3. Specificity digunakan untuk membandingkan proporsi tn terhadap tupel yang negative, yang dihitung dengan menggunakan persamaan:

$$spesificity = \frac{tn}{tn + fp}$$

4. PPV (positive predictive value) atau presicion adalah proporsi kasus dengan hasil diagnosa positif, yang dihitung dengan persamaan :

$$PPV = \frac{tp}{tp + fp}$$

5. NVP (Negative Predictive value) adalah proporsi kasus hasil diagnosa negative yang dihitung dengan persamaan :

$$NVP = \frac{tn}{tn + fn}$$

Selanjutnya menurut Gorunescu 2011 (Wahono, 2012), klasifikasi akurasi dapat di tunjukan pada table berikut ini.

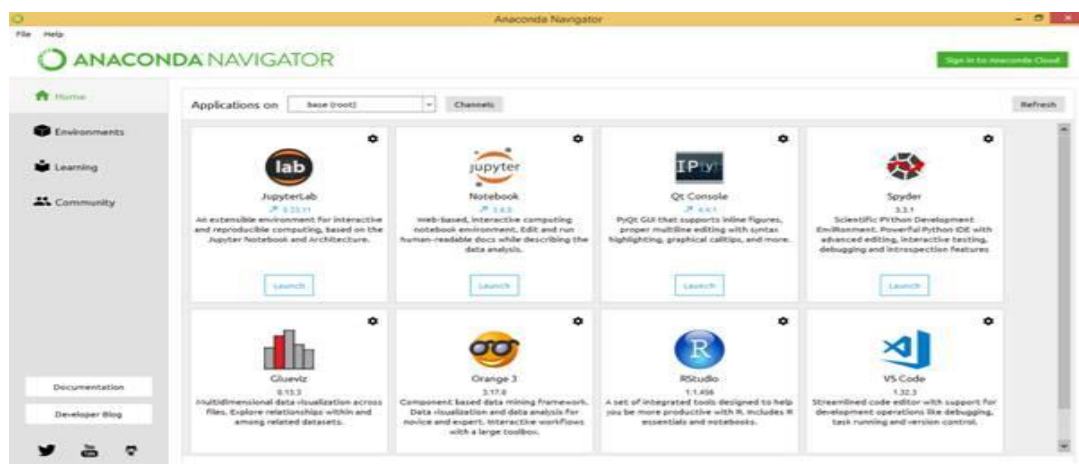
Tabel 2.11. Klasifikasi Akurasi

Akurasi (%)	Klasifikasi
90-100	Exellent
80-89	Good
70-79	Fair
60-69	Poor
50-59	Failure

2.5. Perangkat Lunak Pengembang

Anaconda adalah bahasa pemrograman Python dan R yang terdistribusi bebas dan mempunyai sumber terbuka untuk komputasi ilmiah (ilmu data, aplikasi pembelajaran mesin, pemrosesan data skala besar, analisis prediktif, dll.). Yang bertujuan untuk menyederhanakan paket manajemen dan penyebarannya. Versi paket dikelola oleh paket konda sistem manajemen. Distribusi Anaconda

digunakan oleh lebih dari 12 juta pengguna dan mencakup lebih dari 1400 paket sains data populer yang cocok untuk Windows, Linux, dan MacOS.



Gambar 2.11. Antarmuka Program Aplikasi Pengembangan Anaconda navigator

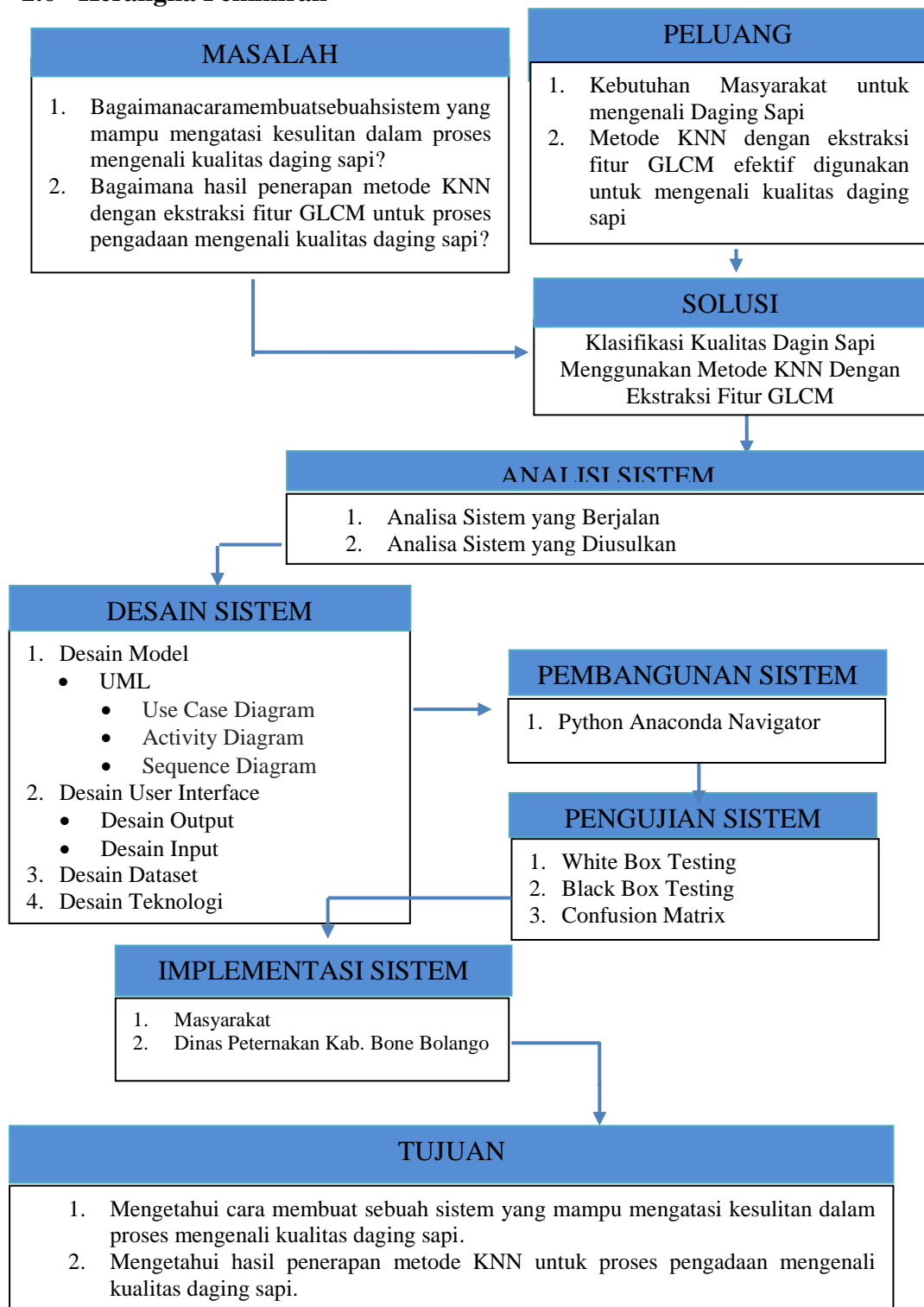
Anaconda Navigator adalah antarmuka pengguna dengan grafis desktop (GUI) yang termasuk dalam distribusi Anaconda. Hal ini memungkinkan kita untuk menjalankan aplikasi yang disediakan dalam distribusi Anaconda dan dengan mudah mengelola paket conda, lingkungan, dan saluran tanpa menggunakan perintah baris-perbari. Ini tersedia untuk Windows, macOS dan Linux.

Aplikasi yang disediakan dalam Distribusi Anaconda :

- JupyterLab
- Jupyter Notebook
- Qt Console
- Spyder
- VS Code
- Rs Studio

- Glueviz
- Orange 3

2.6 Kerangka Pemikiran



Gambar 2.12 Kerangka Pemikiran

BAB III

OBJEK DAN METODE PENELITIAN

3.1. Objek Penelitian

Objek dari penelitian ini adalah “Mengenali Kualitas Daging Sapi”. Penelitian ini bertempat di Dinas Peternakan Kab. Bone Bolango.

3.2 Metode Penelitian

Dalam penelitian ini menggunakan metode penelitian deskriptif, yaitu suatu metode yang menggambarkan suatu keadaan yang sementara berjalan pada saat penelitian dilakukan dan memeriksa sebab-sebab dari suatu gejala tertentu secara sistematis berdasarkan data-data yang ada. Tujuan dari penelitian ini adalah untuk membuat gambaran atau lukisan secara sistematis, aktual, dan akurat mengenai fakta-fakta, sifat-sifat, serta hubungan antar fenomena yang diteliti.

Berdasarkan pengertian tersebut, maka penulis/peneliti menarik kesimpulan bahwa metode analisis deskriptif cocok untuk digunakan dalam penelitian ini, karena sesuai dengan maksud dari penelitian, yaitu untuk memperoleh gambaran tentang “Klasifikasi Kualitas Daging Sapi Menggunakan Metode KNN Dengan Ekstraksi Fitur GLCM”.

3.2.1 Tahap Pengumpulan Data

Untuk mengumpulkan data di gunakan 2(dua) jenis data yaitu data primer dan data sekunder. Data primer yaitu data yang berasal dari penelitian lapangan dan data sekunder berasal dari penelitian kepustakaan.

1. Data Primer

Untuk memperoleh data primer yang merupakan data langsung dari objek penelitian yaitu citra daging sapi segar berkualitas baik, daging gelonggongan dan daging sapi busuk. Adapun Pada penelitian ini dilakukan dengan cara mengumpulkan data citra daging sapi pada 3 jarak yaitu 2cm, 4cm dan 6 cm di tempat terbuka (outdoor) dilakukan menggunakan kamera handphone Xiaomi Note 8 dengan ukuran kamera 48MP. Pada pelatihan data citra diambil 10 data citra untuk masing-masing jarak dengan masing-masing kualitas daging.

2. Data Sekunder

Data sekunder merupakan data yang di peroleh dari studi literatur yang dilakukan. Sumber dari studi literatur yaitu jurnal, makalah ilmiah atau buku yang membahas penelitian tentang kualitas daging sapi menggunakan Metode KNN.

3.2.2 Tahap Analisis Sistem

Tahap analisis merupakan tahap penguraian dari suatu sistem informasi dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dalam kebutuhan - kebutuhan yang di harapkan sehingga dapat diusulkan perbaikan - perbaikannya, di dalam

tahap analisis sistim terdapat langkah-langkah dasar yang harus di lakukan oleh analisis sistem sebagai berikut :

1. *Identify*, yaitu mengidentifikasi masalah.
2. *Understand*, yaitu menganalisa sistem.
3. *Report*, yaitu membuat laporan hasil analisis.

3.2.2.1 Analisis Sistem Berjalan

Analisis sistem berjalan adalah saat ini cara mengenalikualitasdaging sapi masih dilakukan berdasarkan pengamatan visual secara langsung pada daging sapi yang akan dikenali. Kelemahan pengenalan manual sangat dipengaruhi subjektifitas manusia sehingga pada kondisi tertentu tidak konsisten proses pendeteksiannya.

3.2.2.2 Analisis Sistem Usulan

Berdasarkan analisis permasalahan yang di lakukan terhadap sistem yang berjalan pada penelitian ini, maka di usulkan membangun sistemyang dapat menjamin keseragaman kualitas pada daging sapi. Perkembangan teknologi informasi memungkinkan mengenalidaging sapi berdasarkan warna dengan bantuan komputer. Metode pengukuran non-konvensional yaitu dengan menggunakan pengolahan citra (*image processing*) menghasilkan data yang akan diproses dengan Metode KNN kemudian diolah dengan menggunakan perangkat lunak komputer sehingga dapat digunakan untuk menentukan kualitas daging sapi. Penelitian ini bertujuan untuk membangun sebuah program atau perangkat lunak dengan metode Metode KNN yang mampu mengenali daging sapi berkualitas baik berdasarkan citra warna.

3.2.3 Tahap Desain Sistem

Tahap ini merancang sistem yang diusulkan berdasarkan hasil analisis yang telah dilakukan sebelumnya. Tahap ini merupakan strategi untuk memecahkan masalah dan mengembangkan solusi terbaik bagi permasalahan sistem. Jika pada tahap analisis menekankan pada masalah bisnis, maka sebaliknya tahap desain fokus pada sisi teknis dan implementasi perangkat lunak dari sistem yang diusulkan. Tahap desain merupakan tugas dan aktivitas yang difokuskan pada spesifikasi detail dari solusi berbasis komputer. Alat (*tools*) yang digunakan dalam desain sistem ini, dalam hal ini untuk desain model, adalah UML (Unified Modeling Language). Untuk Desain *Output* dan *Input* menggunakan *Ms. Visio*.

Tahap Konstruksi Sistem Tahap konstruksi adalah tahapan menerjemahkan hasil pada tahap desain sistem ke dalam kode-kode program komputer. Pada tahap ini akan Menggunakan Python Anaconda Navigator yang di dalamnya terdapat aplikasi Spyder.

3.2.4 Tahap Pengujian Sistem

Pengujian perangkat lunak, mengukur efisiensi dan efektifitas alur logika pemrograman yang dirancang dengan menggunakan pengujian *White Box Testing* dan *Black Box Testing*. *White Box Texting* menguji perangkat lunak yang telah selesai dirancang kemudian diuji dengan cara: bagan alir program (*flowchart*) yang dirancang sebelumnya dipetakan kedalam bentuk bagan alir kontrol (*flowgraph*) yang tersusun dari beberapa *node* dan *edge*. *Flowgraph* memudahkan penentuan jumlah *region*, *cyclomatic complexity* (CC), dan apabila *independent*

path sama besar, maka sistem dinyatakan benar, tetapi jika sebaliknya, maka sistem masih memiliki kesalahan.

Sedangkan *Black Box Testing* memfokuskan pada keperluan fungsional dari perangkat lunak. *Black Box Testing* bukan merupakan alternatif dari *White Box Testing*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya. *Black Box Testing* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

Fungsi-fungsi yang salah atau hilang

- a. Kesalahan interface
- b. Kesalahan dalam struktur data atau akses basis data eksternal
- c. Kesalahan performa
- d. Kesalahan inisialisasi dan terminasi.

3.2.5 Tahap Pemeliharaan Sistem

Klasifikasi Daging Sapi Menggunakan Metode KNN Dengan Ekstraksi Fitur GLCM telah diimplementasikan kemudian akan dievaluasi kelayakannya dan dilakukan pemeliharaan (*maintenance*) secara berkala baik terjadi kerusakan terhadap sistem maupun tidak.

3.2.6 Tahap Implementasi





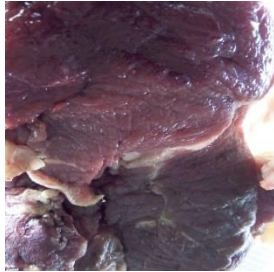
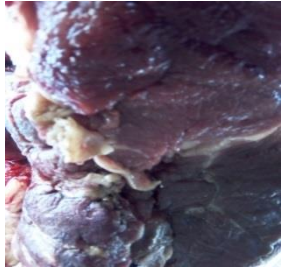
Klasifikasi Kualitas Daging Sapi Menggunakan Metode KNN Dengan Ekstraksi Fitur GLCM ini akan di implementasikan pada Masyarakat yang membutuhkan dan Kantor Dinas Peternakan Kab. Bone Bolango.

BAB IV

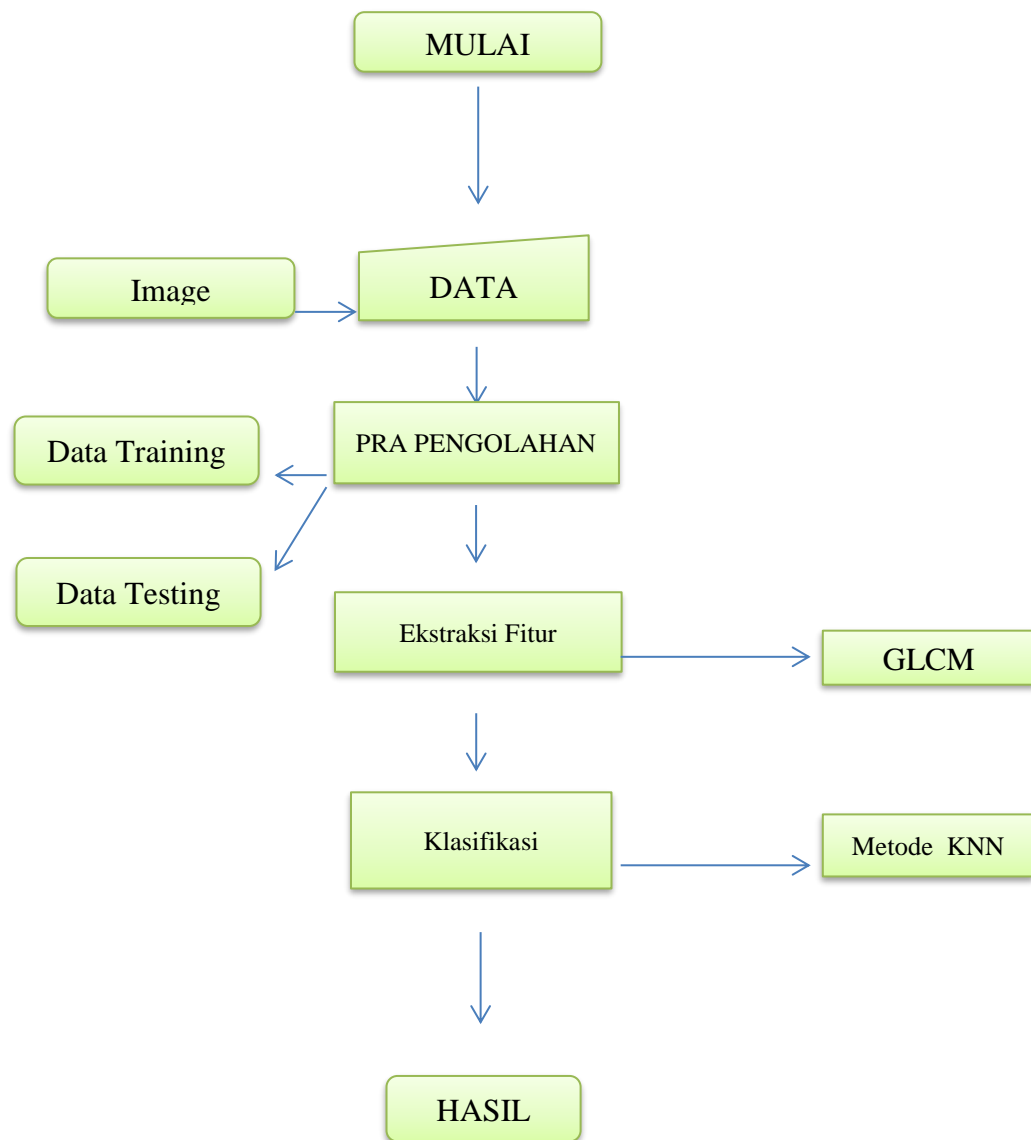
HASIL PENELITIAN

4.1. Hasil Pengumpulan Data

Tabel 4.1 Data Penelitian

Segar			
Tidak segar			

4.2. Arsitektur Metode KNN



Gambar 4.1 Arsitektur

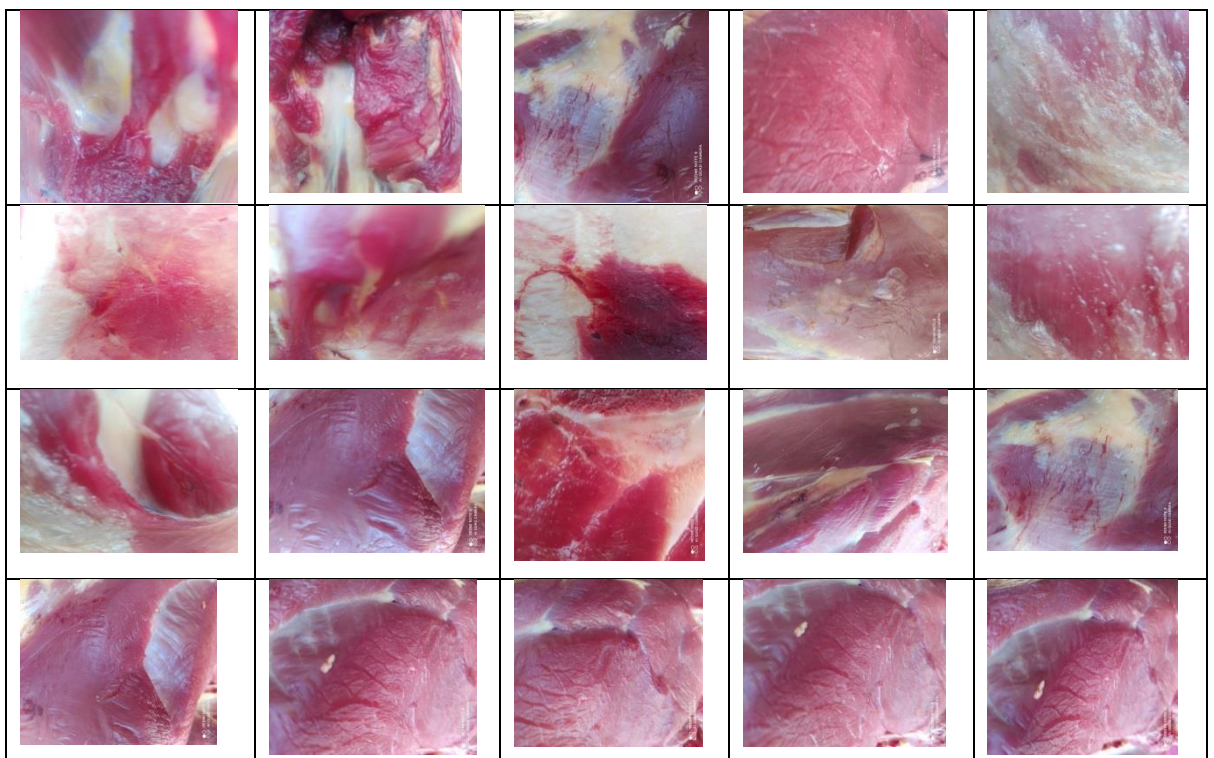
4.3. Penerapan Metode

1. Data Penelitian

a. Pembagian Data Training

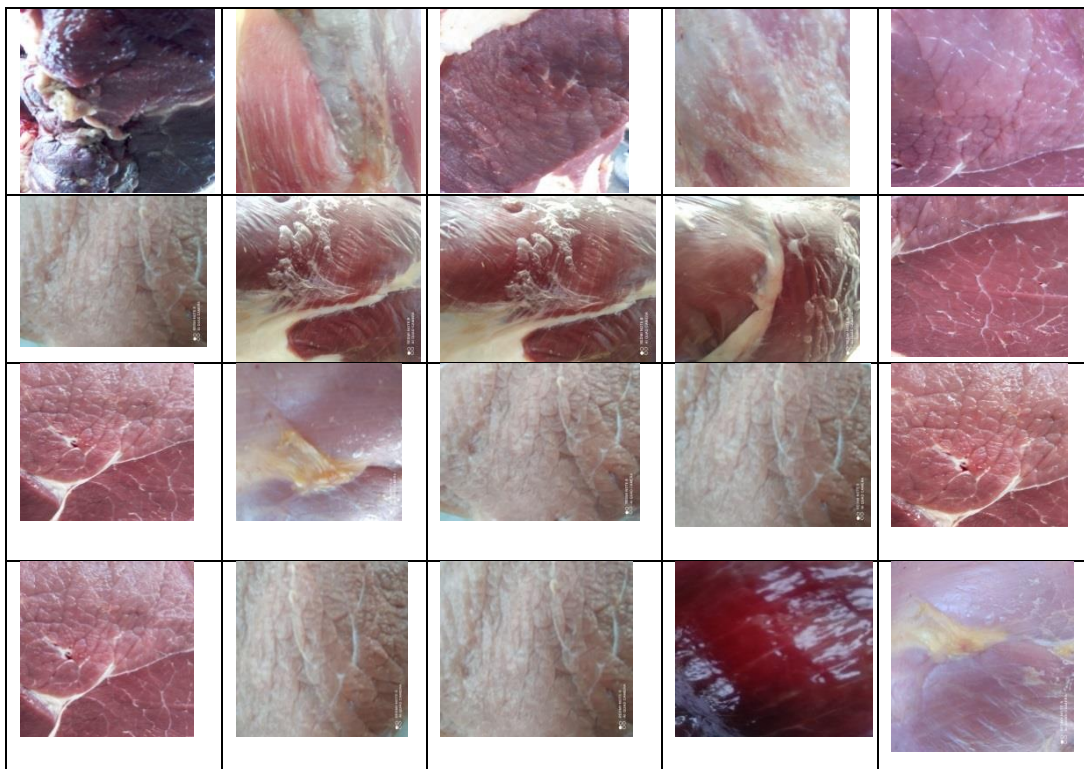
Pembagian data training terbagi dalam 2 kualitas yaitu Segar dan Tidak Segar yang masing-masing memiliki 100 data di tiap kualitas berikut ini.

1. Segar



Gambar 4.2 Pembagian Data Training Segar

2. Tidak Segar

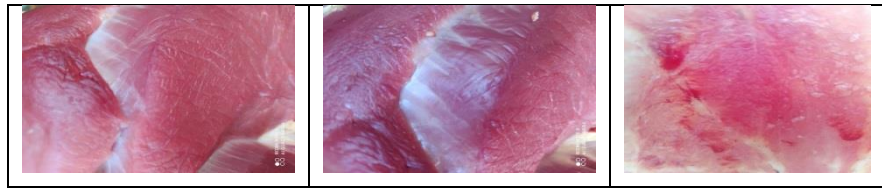


Gambar 4.3 Pembagian DataTraining Daging Sapi Tidak Segar

b. Pembagian Data Testing

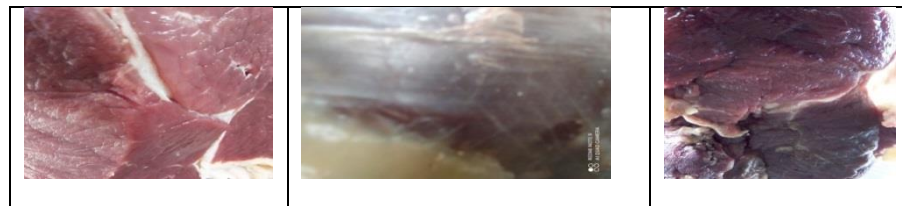
Pada data testing data yang digunakan yaitu data gambar baru di setiap Daging Sapi Segar dan Tidak Segar berikut ini.

1. Segar



Gambar 4.4 Pembagian DataTesting Daging Sapi Segar

2. Tidak Segar



Gambar 4.5 Pembagian DataTesting Daging Sapi Tidak Segar

4.4 Nilai GLCM

Berdasarkan hasil yang diperoleh dari data gambar daging sapi kualitas segar atau tidak segar di atas maka nilai GLCM berdasarkan sudut 0^0 , 45^0 , 90^0 , dan 135^0 untuk pada masing masing metric texture (contrast, energy, homogeneity, dan correlation) adalah sebagai berikut :

Tabel 4.2 Nilai GLCM 0⁰

No/Index		contrast	energy	homogeneity	correlation
0	1	1173.825346733	0.399706258	0.457749270	0.713935049
1	2	948.614069919	0.362189085	0.433848084	0.726916953
2	3	1190.049080645	0.391012939	0.446623558	0.763805220
3	4	728.411112903	0.437145653	0.520808872	0.825664247
4	5	1819.195887096	0.278454544	0.356164783	0.750554058
5	6	1338.763601557	0.218492769	0.307807802	0.782545197
6	7	991.411435483	0.273558444	0.396764827	0.866292901
7	8	1209.465758064	0.432350770	0.491702093	0.718464936
8	9	2160.853596774	0.224590676	0.295799211	0.738867004
99	100	1331.947008907	0.394732900	0.454075566	0.722747513
	
100	1	1513.753370370	0.143905714	0.248131411	0.797136358
101	2	3625.092758064	0.119229666	0.208160551	0.640687329
102	3	662.759323943	0.296533184	0.385909284	0.717181398
103	4	1817.837254549	0.227007756	0.301156712	0.621753047
104	5	940.852593406	0.260350338	0.418230967	0.840049836
105	6	934.113511520	0.223390551	0.344735747	0.534836705
106	7	926.960433604	0.251016274	0.376146424	0.607471834
107	8	1856.951113331	0.302701226	0.379720212	0.662518520

108	9	1725.671671457	0.222352069	0.295311255	0.615029951
199	100	468.553432432	0.273447842	0.449658198	0.821083373

Tabel 4.3 Nilai GLCM 45⁰

No/Index		Contrast	energy	homogeneity	correlation
0	1	1038.631451612	0.406177807	0.479877812	0.746750661
1	2	739.952692822	0.365935806	0.456962230	0.787075035
2	3	1132.700463142	0.390859382	0.444066006	0.775213404
3	4	638.618295763	0.439181823	0.547961597	0.847042641
4	5	1853.034055577	0.276521843	0.355623195	0.746161194
5	6	1506.201375545	0.218432196	0.302328857	0.755762968
6	7	1770.939305609	0.261514066	0.341320115	0.761179113
7	8	1301.180479984	0.425881982	0.478157473	0.696676299
8	9	2170.824345770	0.224372675	0.295979293	0.737379709
99	100	1741.154222952	0.376662871	0.432135603	0.638398742
100	1	2032.832150289	0.139352563	0.211818872	0.727594309
101	2	4171.640416504	0.109760434	0.178693432	0.585647198
102	3	601.1724847732	0.300105136	0.400873924	0.743067914
103	4	2299.122755753	0.212104289	0.268316446	0.520280296
104	5	849.8130618720	0.262599585	0.422178783	0.854952425
105	6	887.600516795	0.226840606	0.344965208	0.558426842

106	7	866.395069913	0.254340415	0.377070804	0.633431302
107	8	2352.958978772	0.284252203	0.340440744	0.572530097
108	9	2195.382606553	0.206611572	0.263486502	0.509121945
199	100	413.9281770946	0.279716012	0.469825341	0.842016045

Tabel 4.4 Nilai GLCM 90⁰

No/Index		Contrast	energy	homogeneity	correlation
0	1	1004.109415322	0.407743226	0.477745448	0.755151444
1	2	964.699948511	0.357049728	0.435968220	0.722409639
2	3	1219.490931467	0.388266149	0.442469872	0.757976384
3	4	822.076710066	0.433546307	0.512694432	0.803088035
4	5	1899.325770825	0.275752282	0.355119230	0.739812182
5	6	1516.625393013	0.217163859	0.310876539	0.754010096
6	7	1217.251894675	0.269209880	0.368026228	0.835830471
7	8	1123.201402383	0.430940089	0.491926307	0.738142986
8	9	2081.159168933	0.225845662	0.298396446	0.748228162
99	100	1796.807502738	0.376373955	0.425511299	0.626874559
100	1	1830.734205440	0.141494862	0.219123372	0.754588075
101	2	4361.94835794	0.109252585	0.175279196	0.566757874
102	3	548.986775321	0.301130274	0.409449730	0.765520332
103	4	2147.969725268	0.217298205	0.275037191	0.551863753

104	5	863.217213114	0.262576076	0.416770201	0.852703605
105	6	930.447323735	0.222100078	0.331519018	0.537106997
106	7	892.597034715	0.252129758	0.367434560	0.622315482
107	8	2414.005701376	0.283500376	0.340213614	0.561424807
108	9	2060.816888396	0.212826807	0.272080922	0.539189693
199	100	451.579914483	0.277704980	0.493282263	0.827783786

Tabel 4.5 Nilai GLCM 135⁰

No/Index		contrast	energy	homogeneity	Correlation
0	1	1173.825346733	0.399706258	0.457749270	0.713935049
1	2	948.614069919	0.362189085	0.433848084	0.726916953
2	3	1190.049080645	0.391012939	0.446623558	0.763805220
3	4	728.411112903	0.437145653	0.520808872	0.825664247
4	5	1819.195887096	0.278454544	0.356164783	0.750554058
5	6	1338.763601557	0.218492769	0.307807802	0.782545197
6	7	991.411435483	0.273558444	0.396764827	0.866292901
7	8	1209.465758064	0.432350770	0.491702093	0.718464936
8	9	2160.853596774	0.224590676	0.295799211	0.738867004
99	100	1331.947008907	0.394732900	0.454075566	0.722747513
	
100	1	1513.753370370	0.143905714	0.248131411	0.797136358
101	2	3625.092758064	0.119229666	0.208160551	0.640687329

102	3	662.759323943	0.296533184	0.385909284	0.717181398
103	4	1817.837254549	0.227007756	0.301156712	0.621753047
104	5	940.852593406	0.260350338	0.418230967	0.840049836
105	6	934.113511520	0.223390551	0.344735747	0.534836705
106	7	926.960433604	0.251016274	0.376146424	0.607471834
107	8	1856.951113331	0.302701226	0.379720212	0.662518520
108	9	1725.671671457	0.222352069	0.295311255	0.615029951
199	100	468.553432432	0.273447842	0.449658198	0.821083373

Dari Nilai GLCM 0^0 , 45^0 , 90^0 , dan 135^0 untuk pada masing masing metric texture(contrast, energy, homogeneity, dan correlation tabel di atas tersebut di dapatkan dengan menjalankan sebuah program dengan cara mendebug File(CTRL+F5) Selanjutnya dengan mengklik Continue execution until next breakpoint(CTRL+F12) Langkah-Langkahnya seperti berikut ini.

Langkah Pertama Kita klik atau breakpoint pada coding GLCM yang ditentukan sehingga muncul lingkaran merah untuk menandakan point dalam mencari nilai GLCM.

```

debugger.py x  __init__.py x  GLCM.py x  form.py x
66     imPath = self.imPaths(imfolder)
67     for item in imPath:
68         img = cv2.imread(item)
69         hasil_preprocessing, grayscale = self.preprocessing_1(img)
70         hasil_ekstraksi = self.ekstraksi_fitur(hasil_preprocessing, distance,
71         data_latih.append(hasil_ekstraksi)
72     return data_latih
73
74     def create_datatesting(self, pathimg, distance, angle):
75         data_testing = []
76         img = cv2.imread(pathimg)
77         hasil_preprocessing, grayscale = self.preprocessing_1(img)
78         plt.imshow(hasil_preprocessing, cmap='gray')
79         plt.show()
80         hasil_ekstraksi = self.ekstraksi_fitur(hasil_preprocessing, distance, ang
81         data_testing.append(hasil_ekstraksi)
82     return data_testing, grayscale, hasil_preprocessing
83
84     def model_knn(self, imfolder, distance, angle):
85         x = self.create_datalatih(imfolder, distance, angle)
86         y = self.getClass(imfolder)
87         clf = KNeighborsClassifier(n_neighbors=25, weights='distance',
88         algorithm='auto', metric='euclidean')
89         clf.fit(x,y)
90         return clf
91
92     def predicts(self, imtest, imfolder, distance, angle):
93         data_test, grayscale, hasil_segment = self.create_datatesting(imtest, dis
94         model = self.model_knn(imfolder, distance, angle)
95         hasil_klasifikasi = model.predict(data_test)
96         return hasil_klasifikasi, grayscale, hasil_segment
97

```

Selanjutnya kita masuk ke coding form lalu klik Debug File lalu pilih continue execution until next breakpoint

```

GLCM.py x  form.py x
- *-
8 15:44:35 2021
Tsa
:
dialog as fdialog
as ttk
ge, ImageTk
:
f,parent):
= parent
)
= tk.Frame(self.parent, width=560, height=49, borderwidth=0
grid(row=0, column=0, padx=10, pady=5, columnspan=2, sticky

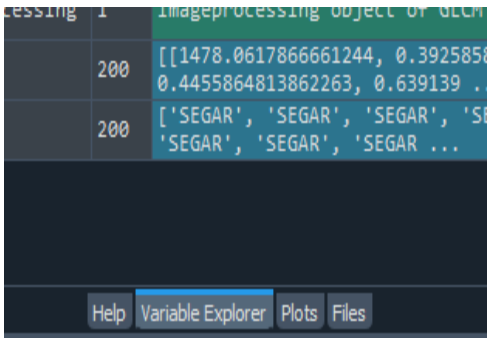
```



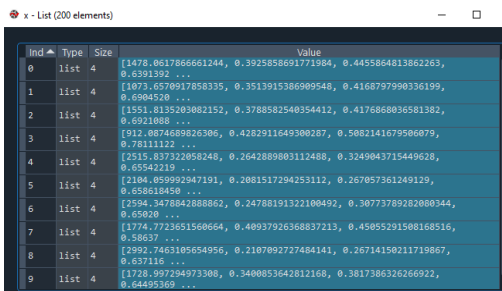
Maka akan muncul tampilan program dan kemudian untuk mencari nilai contrast yaitu dengan memilih gambar mana yang akan kita cari nilai contrast sampai correlation



Lalu menunggu beberapa saat nantinya maka akan muncul nilainya dan untuk melihat nilai dari contrast sampai correlation klik variable explorer seperti gambar di bawah ini.



Lalu klik variable **x** maka akan terlihat urutan nilai GLCM lalu klik Value untuk melihat satu persatu nilai GLCM nya.



0 - List (4 elements)

Ind ▲	Type	Size	Value
0	float64	1	1478.0617866661244
1	float64	1	0.3925858691771984
2	float64	1	0.4455864813862263
3	float64	1	0.6391392817316215

4.5 Klasifikasi KNN

Berdasarkan hasil yang diperoleh dari data Gambar daging sapi kualitas segar dan tidak segar di atas maka Klasifikasi KNN menggunakan Nilai GLCM 0^0 sebagai berikut:

$$\sqrt{(Data Lama) - (Data Baru)^2 + (Data Lama) - (Data Baru)^2}$$

a. Data Lama

Tabel 4.6 Data Gambar

Gambar	Contrast	Energy	Homogeneity	Correlation	Keterangan
Gambar ke 1	1173.825346733	0.399706258	0.457749270	0.713935049	Segar
Gambar ke 2	948.614069919	0.362189085	0.433848084	0.726916953	Segar

Gambar Ke 3	1190.049080645	0.391012939	0.446623558	0.763805220	Segar
Gambar ke 100	1331.947008907	0.394732900	0.454075566	0.722747513	Segar
.....
Gambar ke 1	1513.753370370	0.143905714	0.248131411	0.797136358	Tidak Segar
Gambar ke 2	3625.092758064	0.119229666	0.208160551	0.640687329	Tidak Segar
Gambar ke 3	662.7593239436	0.296533184	0.385909284	0.717181398	Tidak Segar
Gambar ke 100	468.553432432	0.273447842	0.449658198	0.821083373	Tidak Segar

Misalkan di sebuah pasar tradisional menjual daging sapi yang berkualitas Segar dan tidak segar . Kita akan gunakan algoritma KNN untuk melakukan pengujian metode.

Langkah perhitungan :

1. Tent. Parameter K = jumlah banyaknya tetangga terdekat, Misal K=3
2. Hitung jarak antara data baru dan semua data yang ada di data training.Misalnya di gunakan square distance dari jarak antara data baru dengan semua data yang ada di data training.

Tabel 4.7 Menghitung kedekatan

Gambar	Contrast	Energy	Homogeneity	Correlation	Ket	Square distance ke data baru
Gbr ke 1	1173.825346733	0.399706258	0.457749270	0.713935049	Segar	$\sqrt{(1173.825346733 - 1328.206146469)^2 + (0.399706258 - 0.240959347)^2 + (0.457749270 - 0.323628905)^2 + (0.713935049 - 0.738650112)^2}$ $= 23.833,4807$
Gbr ke 2	948.614069919	0.362189085	0.433848084	0.726916953	Segar	$\sqrt{(948.614069919 - 1328.206146469)^2 + (0.362189085 - 0.240959347)^2 + (0.433848084 - 0.323628905)^2 + (0.726916953 - 0.738650112)^2}$ $= 144.090,172$
Gbr ke 3	1190.049080645	0.391012939	0.446623558	0.763805220	Segar	$\sqrt{(1190.049080645 - 1328.206146469)^2 + (0.391012939 - 0.240959347)^2 + (0.446623558 - 0.323628905)^2 + (0.763805220 - 0.738650112)^2}$ $= 19.087,4132$
Gbr ke 100	1331.947008907	0.394732900	0.454075566	0.722747513	Segar	$\sqrt{(1331.947008907 - 1328.206146469)^2 + (0.394732900 - 0.240959347)^2 + (0.454075566 - 0.323628905)^2 + (0.722747513 - 0.738650112)^2}$ $= 14,0349641$
.....	
Gbr ke 1	1513.753370370	0.143905714	0.248131411	0.797136358	Tidak Segar	$\sqrt{(1513.753370370 - 1328.206146469)^2 + (0.143905714 - 0.240959347)^2 + (0.248131411 - 0.323628905)^2 + (0.797136358 - 0.738650112)^2}$ $= 34.427,7908$
Gbr ke 2	3625.092758064	0.119229666	0.208160551	0.640687329	Tidak Segar	$\sqrt{(3625.092758064 - 1328.206146469)^2 + (0.119229666 - 0.240959347)^2 + (0.208160551 - 0.323628905)^2 + (0.640687329 - 0.738650112)^2}$ $= 5.275.688,14$
Gbr ke 3	662.759323943	0.296533184	0.385909284	0.717181398	Tidak Segar	$\sqrt{(662.759323943 - 1328.206146469)^2 + (0.296533184 - 0.240959347)^2 + (0.385909284 - 0.323628905)^2 + (0.717181398 - 0.738650112)^2}$ $= 442.819,802$
Gbr ke 100	468.553432432	0.273447842	0.449658198	0.821083373	Tidak Segar	$\sqrt{(468.553432432 - 1328.206146469)^2 + (0.273447842 - 0.240959347)^2 + (0.449658198 - 0.323628905)^2 + (0.821083373 - 0.738650112)^2}$ $= 739.002,813$

3. Urutkan jarak tersebut dan tentukan Tetangga mana yang terdekat berdasarkan jarak minimum ke- K.

Tabel 4.8 Berdasarkan urutan Ranking Jarak

Urutan(Ranking Jarak)	Square Distance	Ket
3	23.833,4807	Segar
5	144.090,172	Segar
2	19.087,4132	Segar
1	14,0349641	Segar
4	34.427,7908	Tidak Segar
8	5.275.688,14	Tidak Segar
6	442.819,802	Tidak Segar
7	739.002,813	Tidak Segar

4. Menentukan Kategori dari tetangga terdekat misal K=3

Tabel 4.9 Kategori AntarJarak

Urutan(Ranking Jarak)	Square Distance	Ket
1	14,0349641	Segar
2	19.087,4132	Segar
3	23.833,4807	Segar
4	34.427,7908	Tidak Segar
5	144.090,172	Segar
6	442.819,802	Tidak Segar

7	739.002,813	Tidak Segar
8	5.275.688,14	Tidak Segar

5. Berdasarkan perhitungan jarak terdekat pada data uji menggunakan $K=3$, maka di peroleh hasil klasifikasi dengan kategori berkualitas Segar

Berdasarkan hasil yang di peroleh dari data Gambar Daging Sapi kualitas Segar dan Tidak Segar Pada Pembahasan sebelumnya maka menghasilkan Confusion Matriks Sebagai Berikut.

Tabel 4.10 ConfusionMatriks

Jenis Daging Sapi	Segar	Tidak Segar	Total
Segar	TP = 7	FN = 0	7
Tidak Segar	FP = 3	TN = 10	13
Total	10	10	20

Keterangan :

TP : True Positiif

FP : FalsePositif

TN : True Negatif

FN : False Negatif

Penyelesaian :

1. Dik : TP = 7

TN = 10

FP = 3

$$FN = 0$$

Dit :

1. Tingkat ketepatan ?
2. Tingkat keberhasilan ?
3. Tingkat akurasi ?

Penyelesaian :

1. Tingkat ketepatan

Tabel 4.11 Tingkat Ketepatan

Jenis Daging Sapi	Segar	Tidak Segar	Total
Segar	7	0	7
Tidak Segar	3	10	13
Total	10	10	

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

$$\frac{7}{7 + 3} = 0.70$$

2. Tingkat Keberhasilan

Tabel 4.12 Tingkat Keberhasilan

Jenis Daging Sapi	Segar	Tidak Segar	Total
Segar	7	0	7
Tidak Segar	3	10	13
Total	10	10	

$$\frac{T}{TN + FP}$$

$$\frac{10}{10 + 3} = 0.77$$

3. Tingkat Akurasi

Tabel 4.13 Tingkat akurasi

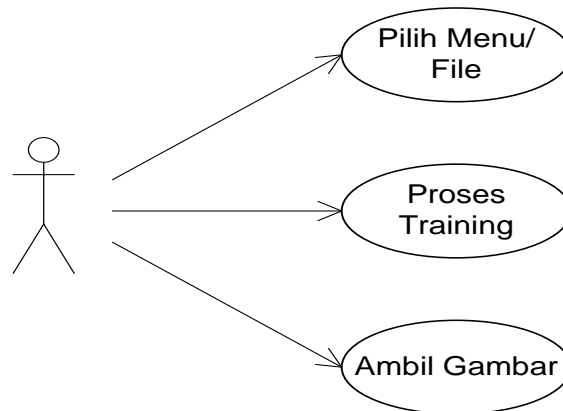
Jenis Daging Sapi	Segar	Tidak Segar	Total
Segar	7	0	7
Tidak Segar	3	10	13
Total	10	10	

$$\frac{TP + TN}{(TP + TN + FP + FN)}$$

$$\frac{7 + 10}{(7 + 10 + 3 + 0)} 0,85\% (85\%)$$

Jadi Hasil Akurasi yang di peroleh dari Jenis Daging Sapi Kualitas Segar dan Tidak Segar berdasarkan data gambar yang di masukkan memiliki tingkat akurasi Sebesar 85%.

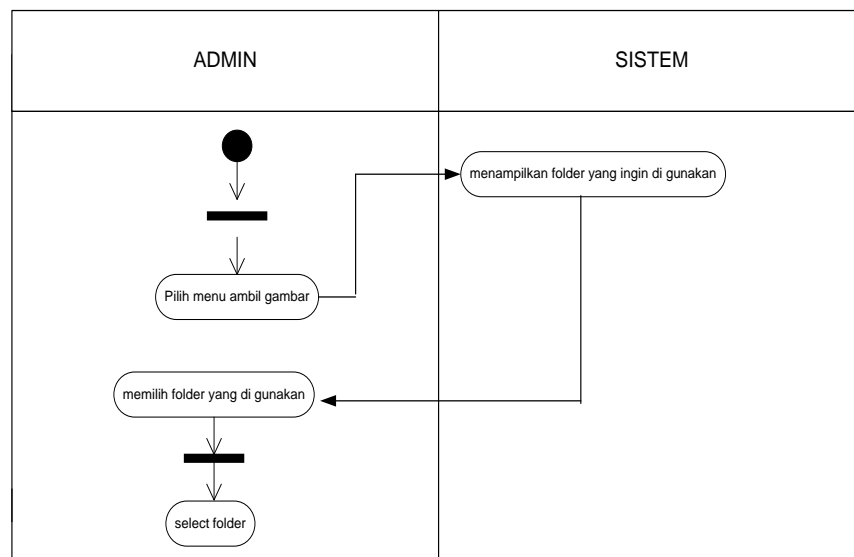
4.6 Sistem yang di usulkan



Gambar 4.6 Sistem yang di usulkan

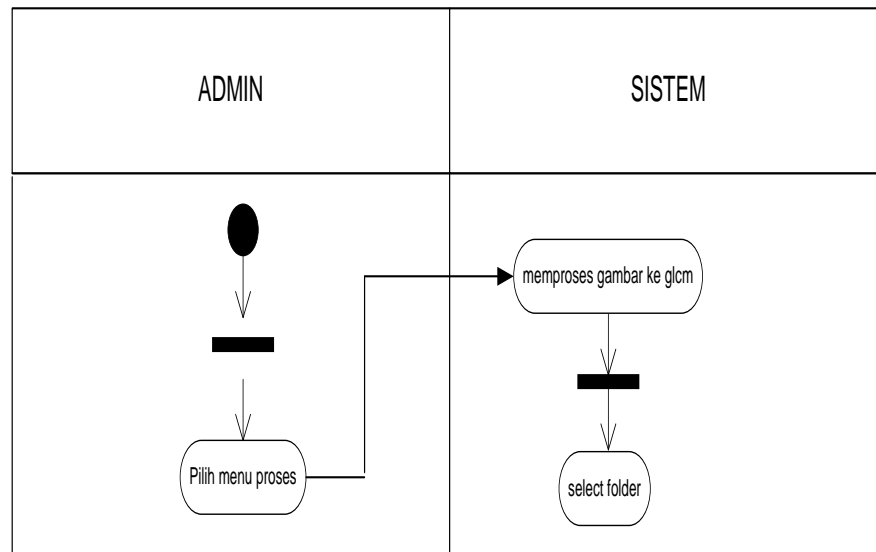
4.7. Activity Diagram

1. Activity diagram Menu pilih file



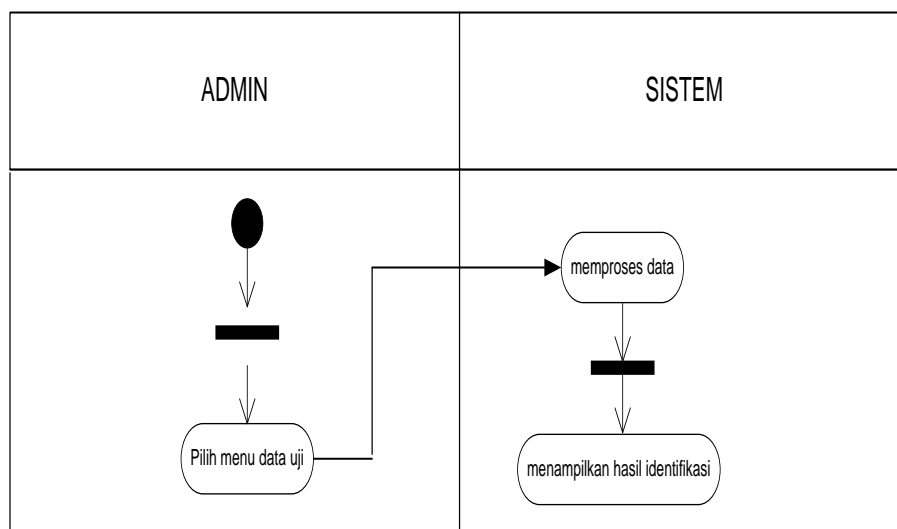
Gambar 4.7 Activity diagram menu pilih file

2. Activity Diagram menu proses glcm



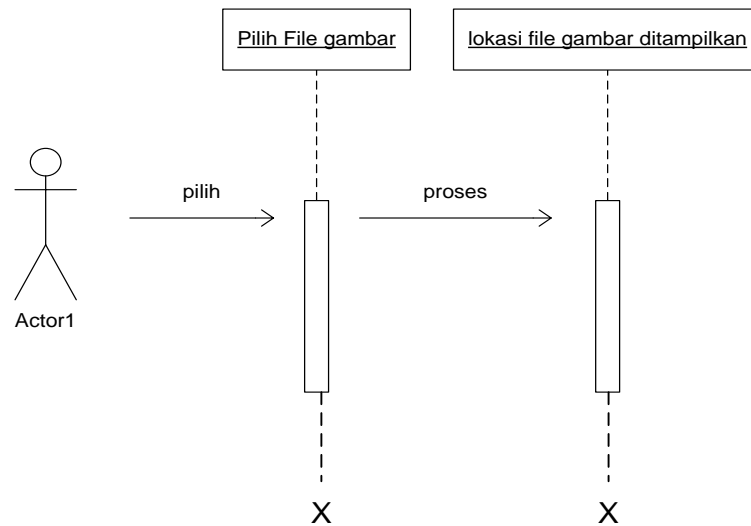
Gambar 4.8 Activity Diagram menu proses

3. Activity Diagram Testing



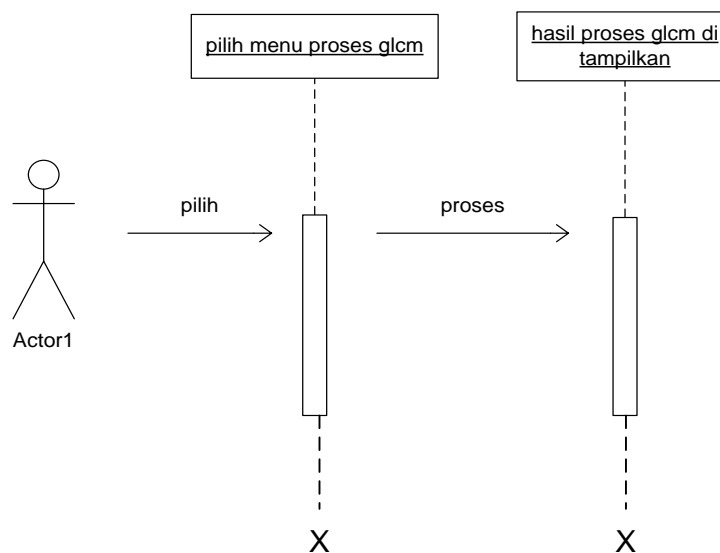
Gambar 4.9 Activity Diagram Testing

4.8. Sequence Diagram pilih file



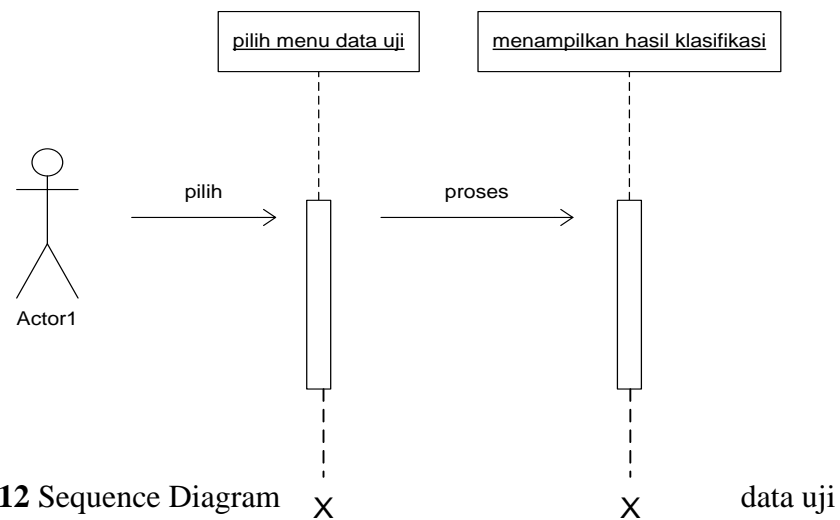
Gambar 4.10 Sequence Diagram Pilih File

4.9. Sequence Diagram Proses GLCM



Gambar 4.11 Sequence Diagram Proses Glcm

4.10. Sequence Diagram data uji



Gambar 4.12 Sequence Diagram X data uji

4.11. Arsitektur Sistem Klasifikasi

Adapun sistem klasifikasi yang di gunakan antara lain :

1. Processor : AMD A4-9125 Radeon R3, 4 Compute Cores 2C +2G
2.30 GHz
2. Memori : 4 GB
3. VGA : Shared Radeon R4 Series
4. Hardisk : 1 TB
5. OS : Windows 10
6. Tools : Python Anaconda Navigator

4.12. Desain Interface

Pendeteksi Daging

NO IMAGE AVAILABLE

citra asli

NO IMAGE AVAILABLE

citra grayscale

NO IMAGE AVAILABLE

citra segmentasi

Hasil

Data Latih **pilih**

Data Uji **pilih**

Arah

jarak 0

PROSES

KLASIFIKASI KUALITAS DAGING SAPI MENGGUNAKAN METODE KNN DENGAN EKTRAKSI FITUR GLCM

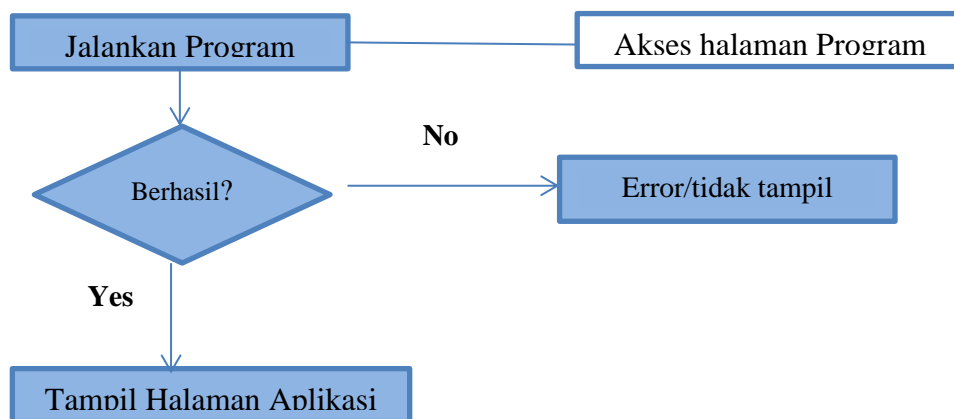
Gambar 4.13 Desain Interface Sistem

4.13 Pengujian White Box

a. Uji coba 1 Pengujian Halaman Aplikasi

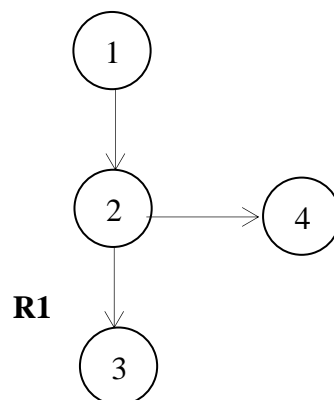
1. Tentukan komponen yang akan di uji

Pada tahapan ini kita tetapkan dulu bagian mana dan apa saja yang ingin kita uji. Buatlah flow pengujian



Gambar 4.13 Flowchart Untuk Pengujian White Box

4.14 Flowgraph Untuk pengujian White Box



Gambar 4.14 Flowgraph Untuk Pengujian White Box

4.15 Perhitungan CC pada pengujian White Box

Dari Flowgraph diatas, maka didapatkan :

Diketahui :

$$\text{Node (N)} = 3$$

$$\text{Edge (E)} = 2$$

$$\text{Predicate Node (P)} = 0$$

$$\text{Region (R)} = 1$$

Rumus :

$$V(G) = (E - N) + 2 \text{ atau}$$

$$V(G) = P + 1$$

Penyelesaian :

$$V(G) = (2 - 3) + 2 = 1$$

$$V(G) = 0 + 1 = 1$$

4.16 Path pada pengujian White Box

Path pada pengujian white box yaitu :

1. 1 - 2 - 3

4.17 Pengujian Aplikasi

Tabel 4.14 Black Box Testing Aplikasi Pendeteksi Daging (Hasil Pengujian Black Box)

No	INPUT/EVENT	FUNGSI	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Saat Pilih Data Latih	Menampilkan folder untuk data latih	Tampil select folder untuk data latih	Sesuai harapan	Valid
2	Saat Pilih Data Uji	Menampilkan Open untuk memilih Gambar data uji	Tampil open untuk pilih gambar Data Uji	Sesuai harapan	Valid
3	Pilih Arah	Menentukan Arah Glcm 0°,45°,90°,dan 135°	Mendapatkan nilai GLCM	Sesuai harapan	Valid
4	Pilih Jarak	Menentukan Jarak 0,2,4,6	Mendapatkan jarak	Sesuai harapan	Valid
5	Proses	Menentukan Hasil Klasifikasi	Mendapatkan hasil klasifikasi Segar atau tidak segar	Sesuai harapan	Valid

BAB V

PEMBAHASAN PENELITIAN

5.1 Hasil Penelitian

5.1.1 Gambaran Umum Lokasi Penelitian

5.1.1.1 Dinas Pertanian Dan Peternakan

Secara makro pembangunan peternakan di Kabupaten Bonebolango selang beberapa tahun terakhir telah menunjukkan perkembangan yang cukup signifikan. Hal ini dapat di lihat antara lain pada kontribusi sub sektor peternakan dari tahun ke tahun cukup besar. Kondisi demikian memberikan indikasi bahwa di masa mendatang pengembangan komoditas peternakan cukup prospektif untuk menjadi komoditas unggulan sekaligus andalan Kabupaten Bonebolango dalam mempercepat laju pembangunan dan kesejahteraan masyarakat.

Berkaitan dengan hal tersebut, maka pemerintah daerah dituntut untuk meningkatkan kualitas pelayanan prima dalam melaksanakan pembangunan di segala bidang. Selanjutnya dalam upaya meningkatkan pelayanan kepada masyarakat, khususnya dalam rangka mewujudkan sarana dan prasarana di bidang peternakan sangat diperlukan suatu perencanaan yang baik dan strategis, sehingga pelaksanaan program dan kegiatan di lapangan dapat terlaksana secara transparan dan terukur.

Sebagai implementasi kewenangan daerah di bidang peternakan, maka Dinas Peternakan dan Kesehatan Hewan Kabupaten Bone Bolango yang di bentuk dengan berpedoman pada peraturan daerah Kabupaten Bonebolango nomor 47

tahun 2014 melaksanakan tugas pokok yaitu melaksanakan kewenangan otonomi daerah dalam rangka pelaksanaan tugas desentralisasi. Dengan demikian kedudukan Dinas Peternakan dan Kesehatan Hewan merupakan unsur pelaksanaan Pemerintah daerah yang dipimpin oleh seorang Kepala Dinas dan bertanggung jawab kepada Bupati Bone Bolango dan mempunyai tugas merencanakan, memimpin dan mengkoordinasikan serta menyelenggarakan seluruh kegiatan di bidang Peternakan dan Kesehatan Hewan.

5.1.1.2 Tugas Pokok dan Fungsi

- **Tugas Pokok**

Dinas mempunyai tugas pokok melaksanakan kewenangan otonomi daerah dalam rangka tugas desentralisasi dan dekonsentrasi di Bidang Peternakan.

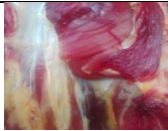
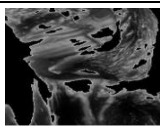

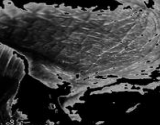

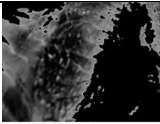

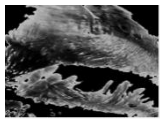

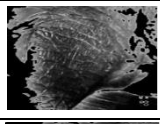
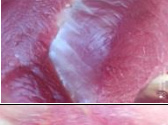
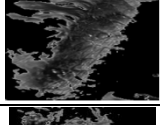

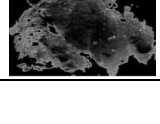
- **Fungsi**




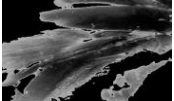



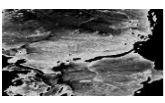

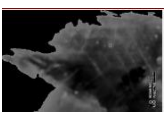

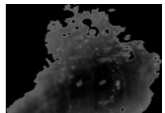

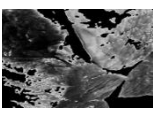
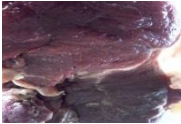
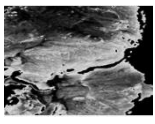
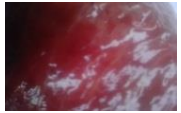


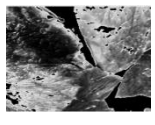

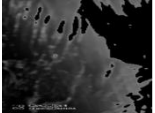
Dalam menyelenggarakan tugas, Dinas mempunyai Fungsi sebagai Berikut:


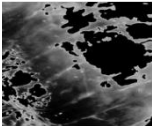
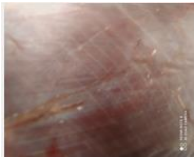
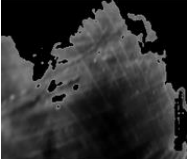
1. Menyusun kebijakan teknis di bidang peternakan.
2. Pelaksanaan pelayanan umum di bidang peternakan.
3. Pembinaan unit pelaksanaan teknis dinas dalam lingkup Dinas Peternakan dan Kesehatan Hewan Kabupaten Bone Bolango

5.1.2 Hasil Pengujian Sistem

Dalam Penelitian ini tahap pengujian system akan dilakukan dengan melakukan pengujian terhadap 20 citra yang ber ekstensi *.jpg. Dari 20 data citra tersebut akan di cari nilai parameternya dengan memilih arah dan jarak sesuai ketentuan program sehingga nantinya akan di dapatkan hasilnya segar atau tidak segar yang nantinya hasil tersebut sesuai atau tidak.

No	Citra Asli	Citra Segment	Manual	Aplikasi	Hasil
1			Segar	Segar	Sesuai
2			Segar	Tidak Segar	Tidak Sesuai
3			Segar	Segar	Sesuai
4			Segar	Tidak Segar	Tidak Sesuai
5			Segar	Segar	Sesuai
6			Segar	Segar	Sesuai
7			Segar	Segar	Sesuai

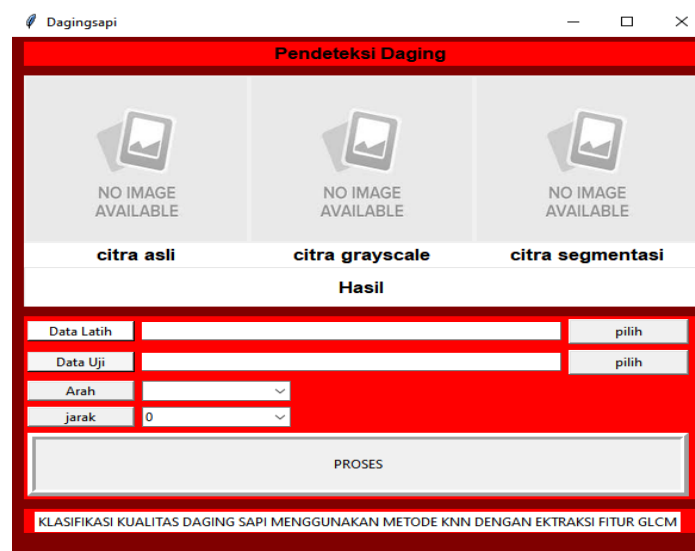
8			Segar	Segar	Sesuai
9			Segar	Segar	Sesuai
10			Segar	Tidak Segar	Tidak Sesuai
11			Tidak Segar	Tidak Segar	Sesuai
12			Tidak Segar	Tidak Segar	Sesuai
13			Tidak Segar	Tidak Segar	Sesuai
14			Tidak Segar	Tidak Segar	Sesuai
15			Tidak Segar	Tidak Segar	Sesuai
16			Tidak Segar	Tidak Segar	Sesuai
17			Tidak Segar	Tidak Segar	Sesuai
18			Tidak Segar	Tidak Segar	Sesuai

19			Tidak Segar	Tidak Segar	Sesuai
20			Tidak Segar	Tidak Segar	Sesuai

Gambar 5.1 Hasil Pengujian Sistem

5.2 Pembahasan Sistem

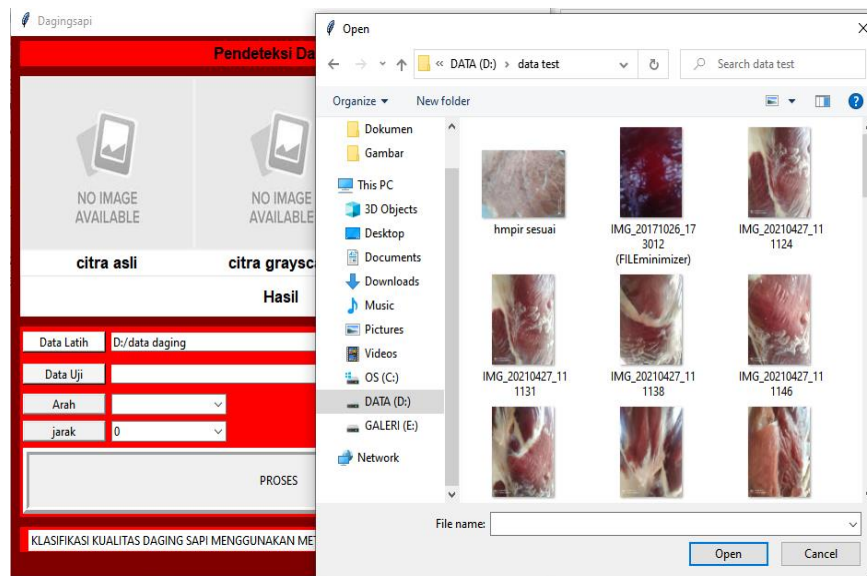
1. Tampilan Menu Utama



Gambar 5.2 Tampilan Menu Utama

Sebelum melakukan pemrosesan sistem akan menampilkan interface Tampilan Menu Utama

2. Tampilan Load Gambar



Gambar 5.3 Tampilan Load Gambar

Pada tampilan ini di jelaskan bagaimana user menginput salah satu gambar.

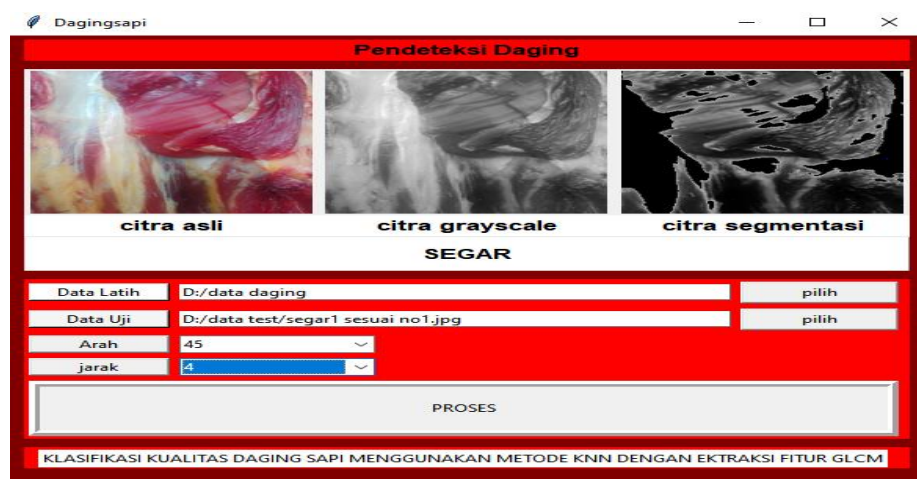
3. Tampilan Menentukan Arah dan Jarak



Gambar 5.4 Tampilan menentukan arah dan jarak

Pada tampilan ini setelah melakukan penginputan gambar ,user memilih arah dan jarak sesuai sistem.

4. Tampilan Hasil Identifikasi



Gambar 5.5 Tampilan Menu Hasil Identifikasi

Pada tampilan ini menunjukkan hasil identifikasi kualitas Segar atau tidak Segar tergantung dari jalannya sistem itu sendiri berdasarkan metode tertentu sehingga menghasilkan akurasi yang di harapkan.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil penelitian yang di lakukan pada Dinas Peternakan dan Kesehatan Hewan Kabupaten Bone Bolango yang telah di uraikan sebelumnya, maka dapat di ambil kesimpulan sebagai berikut :

1. Bahwa aplikasi klasifikasi kualitas daging sapi menggunakan metode KNN dengan ekstraksi fitur GLCM dapat digunakan sebagai alat bantu mengenali kualitas daging sapi
2. Bahwa sistem aplikasi yang di rancang dapat di gunakan. Hal ini dapat di buktikan menggunakan Confusion Matrix. Berdasarkan hasil penelitian yang menggunakan confusion matrix. Confusion matrix merepresentasikan evaluasi yang menghasilkan tingkat ketepatan 70%, tingkat keberhasilan 77% dan nilai akurasi 85% di dapat dari 20 data citra

6.2 Saran

Saran untuk penelitian selanjutnya yang berkaitan dengan penelitian ini,yaitu :

1. Menguji dengan parameter lain pada identifikasi kualitas daging Sapi yang selanjutnya di kombinasikan dengan ekstraksi fitur *Gray Level Co-Occurrence Matrix* (GLCM). sehingga menghasilkan akurasi optimal.
2. Pengolahan citra di lakukan secara realtime agar mendekati kondisi di lapangan.

DAFTAR PUSTAKA

- R. Widodo, Agus Wahyu Widodo, Arry Supriyanto, Pemanfaatan Ciri *Gray Level Co-occurrence Matrix* (GLCM) Citra Buah Jeruk Keprok (*Citrus Reticulata Blanco*) untuk Klasifikasi Mutu, *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vol.2, No.11, November 2018
- H. Wijayanto, Klasifikasi Batik Menggunakan metode K-nearest Neighbour Berdasarkan *Gray Level Co-occurrence Matrix*(GLCM), Teknik Informatika, Unidus, Semarang, 2015
- Hasanah, Uswatun (2018) Klasifikasi Citra Daging sapi menggunakan metode KNN (*K-Nearest Neighbor*) Untuk Identifikasi Kualitas dan Kesegaran. Undergraduate thesis, Universitas Muhammadiyah Sidoarjo.
- Budianita, E., & Handayani, L. 2015. Implementasi Pengolahan Citra dan Klasifikasi K- Nearest Neighbour Untuk Membangun Aplikasi Pembaca Daging Sapi dan Babi, 12(2), 242–247.
- Asmara, R. A., Romario, R., Batubulan, K. S., Rohadi, E., Siradjuddin, I., Ronilaya, F., ... Rahutomo, F. 2018. Classification of pork and beef meat images using extraction of color and texture feature by *Grey Level Co-Occurrence Matrix* method.
- Tarigan, W. 2017. Pengembangan Aplikasi Pendeteksi Daging Celeng Berbasis Android dengan Ekstraksi Fitur Warna dan Klasifikasi KNN.
- Hanang (2015). Klasifikasi Batik Menggunakan Metode K - Nearest Neighbour Berdasarkan *Gray Level CoOccurrence Matrices* (GLCM).
- Dedy, dkk (2015). Aplikasi Mobile Android Pendeteksi Tingkat Kesegaran Daging Sapi Lokal Menggunakan Ekstraksi Fitur Warna Dengan Pendekatan Statistika. *Jurnal Teknik Informatika*. Universitas Dian Nuswantoro Semarang.
- F.Liantoni, dan H. Nugroho, “Klasifikasi Daun Herbal Menggunakan Naïve Bayes Classifier dan K-Nearest Neighbor”, *Jurnal Simantec*, Vol.15, No.1,2015
- D.P. Pamungks , A.B.Setiawan , Implementasi Ekstraksi Fitur dan K-Nearest Neighbor Untuk Identifikasi Wajah Personal, *Joutica* Vol.3 No.2 Lamongan :Unisla, 2018
- Surya, R.A, Fadhil, A & Yudhana, Ekstraksi Ciri Metode *Gray Level Co-occurrence Matrix*(GLCM) dan Filter Galbor Untuk Klasifikasi Citra Batik Pekalongan. *Jurnal Informatika : Jurnal Pengembangan IT*, Volume 02 2017

LISTING PROGRAM

```
# -*- coding: utf-8 -*-
"""
Created on Sat May 8 15:44:35 2021

@author: Mirnawati Isa
"""

1. GLCM

import cv2
from imutils import paths
import os
import numpy as np
from skimage import feature
from sklearn.neighbors import KNeighborsClassifier

from matplotlib import pyplot as plt

class imageprocessing:

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

    def preprocessing_1(self,img):
        grayscale = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        grayscale = cv2.resize(grayscale, (100, 100),
interpolation=cv2.INTER_AREA)
        equalize = cv2.equalizeHist(grayscale)
        blur = cv2.GaussianBlur(grayscale, (5,5), 0)
        ret, alpha = cv2.threshold(blur, 50, 255,
                                cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
        alpha[alpha==255] = 1
        hasil_segment = alpha * equalize
        # kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (9,9))
        # hasil = cv2.dilate(hasil_segment, kernel, iterations=5)
        # kernel_1 = np.ones((5, 5), np.uint8)
        # hasil_1 = cv2.erode(hasil_segment, kernel_1)
        cnts = cv2.findContours(hasil_segment, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
        cnts = cnts[0] if len(cnts) == 2 else cnts[1]
        cnts = sorted(cnts, key=cv2.contourArea, reverse=True)
        for c in cnts:
```

```

        x,y,w,h = cv2.boundingRect(c)
        ROI = hasil_segment[y:y+h, x:x+w]
        break

    return ROI, grayscale

def ekstraksi_fitur(self,img, distance, angel):
    glcm = feature.greycomatrix(img, [distance], [angel], symmetric=False,
                                normed=True)
    con = feature.greycoprops(glcm, 'contrast')[0,0]
    energy = feature.greycoprops(glcm, 'energy')[0,0]
    hom = feature.greycoprops(glcm, 'homogeneity')[0,0]
    cor = feature.greycoprops(glcm, 'correlation')[0,0]
    return [con, energy, hom, cor]

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

def create_datalatih(self,imfolder, distance, angle):
    data_latih = []
    imPath = self.imPaths(imfolder)
    for item in imPath:
        img = cv2.imread(item)
        hasil_preprocessing, grayscale =self.preprocessing_1(img)
        hasil_ekstraksi = self.ekstraksi_fitur(hasil_preprocessing, distance, angle)
        data_latih.append(hasil_ekstraksi)
    return data_latih

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

def model_knn(self,imfolder, distance, angle):
    x = self.create_datalatih(imfolder, distance, angle)
    y = self.getClass(imfolder)
    clf = KNeighborsClassifier(n_neighbors=25, weights='distance',

```

```

        algorithm='auto', metric='euclidean')
    clf.fit(x,y)
    return clf

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

# if __name__ == '__main__':
#     folder_train = "D:/MP2 proposal 1/proposal/SAPi/gambar"
#     folder_test = "D:/MP2 proposal 1/proposal/SAPi/gambar/data sapi.jpg"
#     print (predicts("folder_test, folder_train"))

```

2. TAMPILAN PROGRAM

```

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

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

    def frame(self):
#frame judul
        self.frame1 = tk.Frame(self.parent, width=560, height=40, borderwidth=0,
background="red", relief='ridge')
        self.frame1.grid(row=0, column=0, padx=10, pady=5, columnspan=2,
sticky="NSEW")
        self.frame1.grid_propagate(0)
#frame gambar
        self.frame2 = tk.Frame(self.parent, width=565, height=240, borderwidth=0,
background="red", relief='ridge')
        self.frame2.grid(row=1, column=0, padx=10, pady=5, columnspan=2,
sticky="NSEW")
        self.frame2.grid_propagate(0)
#frame input

```

```

        self.frame3 = tk.Frame(self.parent, width=565, height=190, borderwidth=0,
background="red", relief='ridge')
        self.frame3.grid(row=2, column=0, padx=10, pady=5, columnspan=2,
sticky="NSEW")
        self.frame3.grid_propagate(0)
#frame credit
        self.frame4 = tk.Frame(self.parent, width=565, height=25, borderwidth=0,
background="red", relief='ridge')
        self.frame4.grid(row=3, column=0, padx=10, pady=5, columnspan=2,
sticky="NSEW")
        self.frame4.grid_propagate(0)

    def widget(self):
#data training
#label
        self.labeldatatrain = tk.Label(self.frame3, text='Data Latih', relief='raised',
width=12, background="white")
        self.labeldatatrain.grid(row=0, column=0, padx=3, sticky="W")
#input
        value_datatraining = tk.StringVar(self.frame3, value="")
        self.inputdatatrain = tk.Entry(self.frame3, textvariable=value_datatraining,
relief='raised', width=58)
        self.inputdatatrain.grid(row=0, column=1, padx=3, pady=3, sticky="W")
#button
        self.buttondatatrain = tk.Button(self.frame3, text='pilih', width=13,
command=lambda: value_datatraining.set(fdialog.askdirectory()))
        self.buttondatatrain.grid(row=0, column=2, padx=3, pady=3, sticky="W")
#data testing
#label
        self.labeldatatest = tk.Label(self.frame3, text='Data Uji', relief='raised',
width=12, bg="white smoke")
        self.labeldatatest.grid(row=1, column=0, padx=3, sticky="W")
#input
        value_datatesting = tk.StringVar(self.frame3, value="")
        self.inputdatatest = tk.Entry(self.frame3, textvariable=value_datatesting,
relief='raised', width=58)
        self.inputdatatest.grid(row=1, column=1, padx=3, pady=3, sticky="W")
#button
        self.buttondatatesting = tk.Button(self.frame3, text='pilih', width=13,
command=lambda: value_datatesting.set(fdialog.askopenfilename()))
        self.buttondatatesting.grid(row=1, column=2, padx=3, pady=3, sticky="W")
#frame3
# GLCM
#label
        self.labelarah = tk.Label(self.frame3, text='Arah', relief='raised', width=12)
        self.labelarah.grid(row=3, column=0, padx=3, pady=3, sticky="W")

```

```

#combobox
    self.valdist = tk.StringVar(self.frame3, value="")
    self.cboxarah = ttk.Combobox(self.frame3, textvariable=self.valdist,
width=17, state='readonly')
    self.cboxarah['values']=('0','45','90','135')
    self.cboxarah.current(0)
    self.cboxarah.grid(row=3, column=1, padx=3, pady=3, sticky="W")
#label
    self.labeljarak = tk.Label(self.frame3, text='jarak', relief='raised', width=12)
    self.labeljarak.grid(row=4, column=0, padx=3, pady=3, sticky="W")
#combobox2
    self.valdist = tk.StringVar(self.frame3, value="")
    self.cboxjarak = ttk.Combobox(self.frame3, textvariable=self.valdist,
width=17, state='readonly')
    self.cboxjarak['values']=('0','2','4','6')
    self.cboxjarak.current(0)
    self.cboxjarak.grid(row=4, column=1, padx=3, pady=3, sticky="W")

#button
    self.buttonproses = tk.Button(self.frame3, text='PROSES', width=29,
height=3, command=self.proses)
    self.buttonproses.grid(row=5, column=0, columnspan=3, padx=3, pady=3,
sticky="NSEW")
    self.buttonproses.configure(borderwidth=7, relief="ridge", anchor="center")

#credit
    self.labeltext3 = tk.Label(self.frame4, text="KLASIFIKASI KUALITAS
DAGING SAPI MENGGUNAKAN METODE KNN DENGAN EKTRAKSI
FITUR GLCM", background="red")
    self.labeltext3.grid(row=1, column=0, padx=9, pady=3)

#judul
    self.labeltext2 = tk.Label(self.frame1, text="Pendeteksi Daging", font="arial
12 bold", background="red", anchor="center")
    self.labeltext2.grid(row=0, column=0, padx=9, pady=3)
    self.labeltext2.pack()

#Frame gambar
#
def hasil(self):
    img = Image.open("gbr.jpg")
    img = img.resize((185,170), Image.ANTIALIAS)
    img = ImageTk.PhotoImage(img)

```

```

self.panel1 = tk.Label(self.frame2, image=img)
self.panel1.image_names = img
self.panel1.grid(row=1, column=0, rowspan=5, sticky="NSEW")
self.labelpanel1 = tk.Label(self.frame2, text="citra asli", font="arial 13
bold", bg="white")
self.labelpanel1.grid(row=7, rowspan=2, column=0, sticky="NSEW")

self.panel2 = tk.Label(self.frame2, image=img)
self.panel2.image_names = img
self.panel2.grid(row=1, column=1, rowspan=5, sticky="NSEW")
self.labelpanel2 = tk.Label(self.frame2, text="citra grayscale", font="arial 13
bold", bg="white")
self.labelpanel2.grid(row=7, rowspan=2, column=1, sticky="NSEW")

self.panel3 = tk.Label(self.frame2, image=img)
self.panel3.image_names = img
self.panel3.grid(row=1, column=2, rowspan=5, sticky="NSEW")
self.labelpanel3 = tk.Label(self.frame2, text="citra segmentasi", font="arial
13 bold", bg="white")
self.labelpanel3.grid(row=7, rowspan=2, column=2, sticky="NSEW")

#Hasil
self.labelhasil = tk.Label(self.frame2, text="Hasil", font="arial 12 bold",
relief="raised", height=2, width=56, background="white")
self.labelhasil.grid(row=9, columnspan=3, column=0, sticky="W")
#frame4

#

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

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

    gambargray = Image.fromarray(grayscale)

```

```

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

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

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

```

```

def main():
    root = tk.Tk()
    root.geometry('585x540+50+50')
    root.configure(background='#800000')
    app = form(root)
    app.parent.title('Dagingsapi')
    root.mainloop()

if __name__=='__main__':
    main()

```




PEMERINTAH KABUPATEN BONE BOLANGO
DINAS PERTANIAN DAN PETERNAKAN

Jl. Lihibadu Desa Bulotalangi Timur (Komp SMP Negeri Bulango Timur)

SURAT KETERANGAN

Nomor : 520/DPP-SEKRET-BB/ 957 /IX/2017

Yang bertanda dibawah ini :

Nama : **Ir. Hi. NURDIN SAUD**
 NIP : 19590305 199103 1 004
 Jabatan : Kepala Dinas Pertanian dan Peternakan Kabupaten Bone Bolango

Dengan ini menerangkan

Nama : **TRISNO**
 NIM : T3113205
 Program Studi : S-1 Ilmu Komputer / Teknik Informatika
 Angkatan :

Adalah benar-benar telah melaksanakan Pengambilan Data Penelitian dalam rangka Skripsi di Dinas Pertanian dan Peternakan Kabupaten Bone Bolango dalam menyelesaikan Studi S-1 Universitas ICHSAN Gorontalo dengan judul Skripsi *"Sistem Cerdas Mengenali Kualitas Daging Sapi Menggunakan Metode Transformasi Wavalet Haar Kabupaten Bone Bolango"*.

Demikian Surat Keterangan ini di berikan kepada yang bersangkutan untuk di gunakan seperlunya.



Kepala Dinas

Ir. Hi. NURDIN SAUD

Nip. 19590305 199103 1 004



**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER**

SK MENDIKNAS NOMOR 84/D/O/2001

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

Berita Acara Perbaikan/Revisi Ujian SKRIPSI

Pada hari ini, Sabtu 05-Juni-2021, Pukul 14.30-16.30 Wita. Telah dilaksanakan Ujian SKRIPSI mahasiswa Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Nama : Trisno
Nim : T3113205
Pembimbing I : Yasin Aril Mustofa, M.Kom
Pembimbing II : Sunarto Taliki, M.Kom
Judul SKRIPSI : Klasifikasi Kualitas Daging Sapi Menggunakan Metode KNN dengan Ekstraksi Fitur GLCM

Oleh Komite Seminar sebagai berikut :

No	Komite Seminar	Status	Tanda Tangan
1	Amiruddin, M.Kom	Ketua	
2	Asmaul Husna, M.Kom	Anggota	
3	Husdi, M.Kom	Anggota	
4	Yasin Aril Mustofa, M.Kom	Anggota	
5	Sunarto Taliki, M.Kom	Anggota	

DAFTAR RIWAYAT HIDUP

Penulis dilahirkan di Marowo pada 20 Oktober 1993 Anak Ketiga dari Tiga Bersaudara dari pasangan Djasno Dj Hasan dan Ruqiah Abubakar



A. IDENTITAS

Nama	: Trisno
Nim	: T3113205
Tempat tanggal lahir	: Marowo, 20 Oktober 1993
Jenis Kelamin	: Laki-laki
Agama	: Islam
Alamat	: Desa Uabanga Kec. Bonepantai
Angkatan	: 2013
Fakultas	: Ilmu komputer
Jurusan	: Teknik Informatika

B. PENDIDIKAN

1. Lulusan SDN 23 Ampana Kota Tahun 2006
2. Lulusan SMPN 2 Ampana Kota Tahun 2009
3. Lulusan SMAN 1 Ampana Kota Tahun 2012
4. Studi Tingkat Strata 1 (S1) Universitas Ichsan Gorontalo
Fakultas Ilmu Komputer Tahun 2013

ABSTRACT**TRISNO. T3113205. THE QUALITY CLASSIFICATION OF BEEF USING K-NEAREST NEIGHBOR METHOD WITH GRAY LEVEL CO-OCCURRENCE MATRIX FEATURE EXTRACTION**

*This study aims to create a system that is able to overcome difficulties in the process of recognizing beef quality and knowing the results of applying the KNN method with GLCM feature extraction for the procurement process to recognize beef quality. This research uses the descriptive research method, which is a method that describes a situation that is currently taking events at the time the research is conducted and examines the causes of a certain symptom systematically based on the existing data. In this study, the system testing phase will be carried out by testing 20 test image data with the *.jpg extension. From the 20 test image data, the parameter values will be investigated by selecting the direction and distance and determined based on 00, 450, 900, and 1350. Thus, based on the calculation of the closest distance to the test data using $K = 3$, then the classification results are obtained with categories Fresh quality and accuracy calculation results obtained from fresh and not fresh beef through the Confusion Matrix calculation which produces an accuracy rate of 0.85% (85%).*

Keywords: Implementation Results, Feature Extraction, KNN Method, GLCM



ABSTRAK

TRISNO. T3113205. KLASIFIKASI KUALITAS DAGING SAPI MENGGUNAKAN METODE *K-NEAREST NEIGHBOR* DENGAN EKSTRAKSI FITUR *GRAY LEVEL CO-OCCURRENCE MATRIX*

Penelitian ini bertujuan untuk membuat sebuah sistem yang mampu mengatasi kesulitan dalam proses mengenali kualitas daging sapi dan mengetahui hasil penerapan metode KNN dengan ekstraksi fitur GLCM untuk proses pengadanan mengenali kualitas daging sapi. Dalam penelitian ini menggunakan metode penelitian deskriptif, yaitu suatu metode yang menggambarkan suatu keadaan yang sementara berjalan pada saat penelitian dilakukan dan memeriksa sebab-sebab dari suatu gejala tertentu secara sistematis berdasarkan data-data yang ada. Dalam Penelitian ini tahap pengujian sistem akan dilakukan dengan melakukan pengujian terhadap 20 data citra uji yang ber ekstensi *.jpg. Dari 20 data citra uji tersebut akan dicari nilai parameternya dengan memilih arah dan jarak, di tentukan berdasarkan 0° , 45° , 90° , dan 135° . Dengan demikian, berdasarkan perhitungan jarak terdekat pada data uji menggunakan $K=3$, maka diperoleh hasil klasifikasi dengan kategori berkualitas Segar dan hasil perhitungan akurasi yang diperoleh dari daging sapi segar dan tidak segar melalui perhitungan Confusion Matrix di dapat tingkat akurasi sebesar 0,85% (85%).

Kata Kunci :Hasil Penerapan, Ekstraksi Fitur, Metode *KNN*, *GLCM*.

