

**DETEKSI KESEGERAN DAGING AYAM *BROILER*
MENGUNAKAN TEKSTUR WARNA DENGAN
METODE *GRAY LEVEL CO-OCCURRENCE*
MATRIX DAN *K-NEAREST NEIGHBOR***

Oleh

ANDI MOH. ILHAM ZULKARNAIN RAHMAN

T3116042

SKRIPSI

Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar Sarjana



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

PERSETUJUAN SKRIPSI

DETEKSI KESEGERAN DAGING AYAM *BROILER* MENGUNAKAN TEKSTUR WARNA DENGAN METODE *GRAY LEVEL CO-OCCURRENCE* *MATRIX* DAN *K-NEAREST NEIGHBOR*

Oleh

ANDI MOH. ILHAM ZULKARNAIN RAHMAN

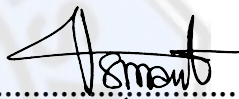
T3116042

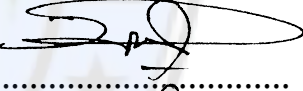
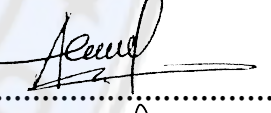
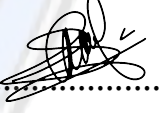
Diperiksa oleh Panitia Ujian Strata Satu (S1)

Universitas Ichsan Gorontalo

Gorontalo, 06 Agustus 2020

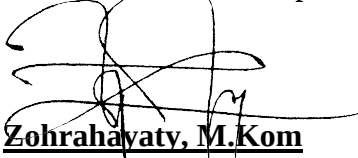
1. Pembimbing I
Asmaul Husna, M.Kom
2. Pembimbing II
Sudirman S. Panna, M.Kom
3. Penguji I
Irvan Abraham Salihi, M.Kom
4. Penguji II
Apriyanto Alhamad, M.Kom
5. Penguji III
Andi Bode, M.Kom


.....

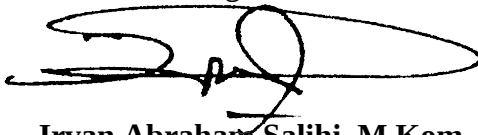
.....

.....

.....

.....

Mengetahui :

Dekan Fakultas Ilmu Komputer


Zohrahayaty, M.Kom
NIDN. 0912117702

Ketua Program Studi


Irvan Abraham Salihi, M.Kom
NIDN. 0928028101

PENGESAHAN SKRIPSI

DETEKSI KESEGERAN DAGING AYAM *BROILER* MENGUNAKAN TEKSTUR WARNA DENGAN METODE *GRAY LEVEL CO-OCCURRENCE* *MATRIX* DAN *K-NEAREST NEIGHBOR*

Oleh

ANDI MOH. ILHAM ZULKARNAIN RAHMAN

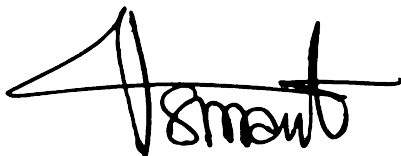
T3116042

SKRIPSI

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

Gorontalo, 15 Juli 2020

Pembimbing I



Asmaul Husna, M.Kom
NIDN. 0911108602

Pembimbing II



Sudirman S. Panna, M.Kom
NIDN. 0924038205

PERNYATAAN SKRIPSI

Dengan ini saya menyatakan bahwa:

1. Karya tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorontalo maupun diperguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari Tim Pembimbing.
3. Dalam karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah dipublikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pula dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, 15 Juli 2020
Yang Membuat Pernyataan,

Andi Moh. Ilham Zulkarnain Rahman

ABSTRACT

Broiler Chicken meat is one type of food that has high economic value and is quite popular with many people. In its use, information on the level of freshness of broiler chicken meat using color texture is very much needed by the livestock industry. But to find out the freshness detection of broiler chicken meat using color texture is still done manually so that the time required is relatively long if done in large quantities. Determination of freshness detection of broiler chicken meat using color texture needs to be done with more [accurate, reliable, efficient, effective, fast, or optimal, etc.] in order to get high accuracy values. Determination of freshness detection of broiler chicken using color texture can be completed using the Gray Level Co-Occurrence Matrix method and K-Nearest Neighbor as its classification model that has never been tested before but is proven to be able and reliable in solving problems such as determining the detection of freshness of broiler chicken meat using color textures. Research results show that the accuracy of the K-NN model for determining the detection of freshness of broiler chicken meat uses 90% color texture using testing. In the K-NN model experiment, this study uses 10 testing data and 90 training data, namely fresh, freezer and rotten categories. with 30 data each. Thus, the application of the Gray Level Co-Occurrence Matrix and K-Nearest Neighbor methods in the above problem managed to get a high accuracy value of 90% by using the Confusion Matrix test.

Keywords: Broiler Chicken Meat Using Color Texture, Application, Gray Level Co-OccurrenceMatrix, K-Nearest Neighbor.

ABSTRAK

Daging Ayam *Broiler* salah satu jenis makanan yang memiliki nilai ekonomis tinggi dan cukup banyak digemari oleh masyarakat. Dalam penggunaannya tentu informasi tingkat kesegaran daging ayam *broiler* menggunakan tekstur warna sangat diperlukan oleh industri peternakan. Namun untuk mengetahui deteksi kesegaran daging ayam *broiler* menggunakan tekstur warna masih dilakukan secara manual sehingga waktu yang dibutuhkan relatif lama jika dilakukan dengan jumlah yang banyak. Penentuan deteksi kesegaran daging ayam *broiler* menggunakan tekstur warna perlu dilakukan dengan lebih [akurat, handal, efisien, efektif, cepat, atau optimal, dll] agar di dapatkan nilai akurasi yang tinggi. Penentuan deteksi kesegaran daging ayam *broiler* menggunakan tekstur warna dapat diselesaikan dengan menggunakan metode *Gray Level Co-Occurrence Matrix* dan *K-Nearest Neighbor* sebagai model Klasifikasinya yang belum pernah di uji coba sebelumnya namun terbukti dapat dan handal dalam menyelesaikan masalah seperti penentuan deteksi kesegaran daging ayam *broiler* menggunakan tekstur warna. Hasil penelitian menunjukkan bahwa akurasi model *K-NN* untuk penentuan deteksi kesegaran daging ayam *broiler* menggunakan tekstur warna sebesar 90% dengan menggunakan pengujian Pada percobaan model *K-NN*, penelitian ini menggunakan 10 data testing dan 90 data training, yaitu kategori segar, freezer dan busuk dengan masing-masing 30 data. Dengan demikian, Penerapan metode *Gray Level Co-Occurrence Matrix* dan *K-Nearest Neighbor* pada masalah diatas berhasil mendapatkan nilai akurasi yang tinggi sebesar 90% dengan menggunakan pengujian *Confusion Matrix*.

Kata Kunci : Daging Ayam *Broiler* Menggunakan Tekstur Warna, Penerapan, *Gray Level Co-OccurrenceMatrix*, *K- Nearest Neighbor*.

KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan skripsi ini dengan judul: “Deteksi kesegaran daging ayam broiler menggunakan tekstur warna dengan metode gray level co-occurrence matrix dan k-nearest neighbor”, untuk memenuhi salah satu syarat penyusunan Skripsi Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penulis menyadari sepenuhnya bahwa skripsi ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materil. Untuk itu, dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan terima kasih banyak dan Bersujud Syukur Kepada Allah S.W.T dan penghargaan yang setinggi-tingginya kepada:

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

9. Bapak dan ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis;
10. Kedua Orang Tua saya, Bapak Hi. Abd. Rahman Adam, SE dan Ibu Sri G. Wahyuni Ibrahim, SE, yang mencintai, atas segala kasih sayang, jerih payah dan doa restunya dalam membesarkan dan mendidik penulis;
11. Rekan-rekan seperjuangan yang telah banyak memberi bantuan dan dukungan moril yang sangat besar kepada penulis;
12. Untuk Pacar yang bernama Hartati Ibrahim Wantu, S.Pd, terima kasih banyak atas doa, dukungan dan bantuannya selama ini untuk penulis;
13. Kepada semua pihak yang ikut membantu dalam penyelesaian skripsi ini yang tak sempat penulis sebutkan satu-persatu.

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

Gorontalo, Juli 2020

Penulis

DAFTAR ISI

HALAMAN SAMPUL	
HALAMAN JUDUL	
PERSETUJUAN SKRIPSI	i
PENGESAHAN SKRIPSI	ii
PERNYATAAN SKRIPSI	iii
ABSTRACT.....	iv
ABSTRAK.....	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN.....	xiii
BAB I PENDAHULUAN.....	1
1. 1 Latar Belakang	1
1. 2 Identifikasi Masalah	5
1. 3 Rumusan Masalah	5
1. 4 Tujuan Penelitian.....	5
1. 5 Manfaat Penelitian.....	5
1.5.1 Manfaat Teoritis	5
1.5.2 Manfaat Praktis	5
BAB II LANDASAN TEORI.....	7
2.1 Tinjauan Studi	7
2.2 Tinjauan Pustaka	8
2.2.1 Analisis Tekstur Warna	8
2.2.2.1 Ekstrasi Ciri Warna	8
2.2.2.2 Ekstrasi Ciri Tekstur.....	9
2.2.2.3 Ekstraksi Ciri Warna	9
2.2.2.4 Ekstraksi Ciri Tekstur.....	9
2.2.2 Computer Vision	10

2.2.3 GLCM (Gray Level Co-Occurrence Matrix)	10
2.2.4 K-NN (K-Nearest Neighbor).....	18
2.3 Kerangka Pikir.....	24
BAB III METODE PENELITIAN	25
3. 1. Jenis, Metode, Subjek, Objek, Waktu Dan Lokasi Penelitian	25
3. 2. Pemodelan.....	26
3. 2. 1. Pra Pengolahan.....	27
3. 2. 2. Ekstraksi Ciri.....	27
3. 2. 3. Data Traning.....	27
3. 2. 4. Traning Menggunakan K-NN	28
3. 2. 5. Model	28
3. 2. 6. Data Testing	28
3. 2. 7. Hasil Klasifikasi	28
3. 2. 8. Evaluasi	29
BAB IV HASIL PENELITIAN	30
4. 1. Hasil Pengumpulan Data.....	30
4. 2. Hasil Pemodelan.....	37
4. 2. 1. Pra Pengolahan.....	37
4. 2. 2. Ekstraksi Fitur	40
4. 2. 3. Klasifikasi.....	44
4. 3. Evaluasi	49
4. 3. 1. Confusion Matrix	49
BAB V PEMBAHASAN	59
4. 4. Pembahasan Model	59
4. 5. Pembahasan Sistem.....	60
4. 6. Hasil Klasifikasi	62
4. 7. Evaluasi	64
BAB VI PENUTUP	66
4. 8. Kesimpulan.....	66
4. 9. Saran.....	66
DAFTAR PUSTAKA	

KODE PROGRAM.....	
1. Pembuatan Form	
2. Image Processing	
DAFTAR RIWAYAT HIDUP	
DATASET (<i>TRAINING & TESTING</i>).....	

DAFTAR GAMBAR

Gambar 2.2.3 Delapan arah sudut untuk GLCM	11
Gambar 2.2.3 Arah dan Jarak dalam GLCM	15
Gambar 2.2.4 Ilustrasi Metode K-NN	20
Gambar 2.2.4 Ilustrasi Aturan Pemilihan KNN	22
Gambar 2.3 Pemodelan / Abstraksi	24
Gambar 3.2 Pemodelan Dataset.....	26
Gambar 4.1 Perbedaan Kesegaran Daging Ayam	30
Gambar 4.5 Tampilan Menu Utama	60

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait.....	7
Tabel 4.1 Mengkonversi RGB ke Grayscale	31
Tabel 4.1 Citra Dengan Image 4x4.....	31
Tabel 4.1 Perhitungan Dengan Langkah Dan Rumus.....	33
Tabel 4.1 Citra 4x4 Asli.....	33
Tabel 4.1 Data Citra 4x4 Setelah Dilakukan Histogram Equalisation.....	33
Tabel 4.2.1 Proses Perubahan Citra Asli Menjadi <i>Grayscale</i>	38
Tabel 4.2.1 Proses Normalisasi Citra.....	39
Tabel 4.2.2 Nilai Dari Suatu Citra	40
Tabel 4.2.2 Matrix GLCM.....	41
Tabel 4.2.2 Matrix Ternormalisasi	41
Tabel 4.2.3 Hasil Perhitungan Data Training	44
Tabel 4.2.3 Hasil Perhitungan Data Testing	46
Tabel 4.2.3 Perbandingan Nilai	46
Tabel 4.2.3 Hasil Pengurutan Dari Nilai Terkecil Berdasarkan Perhitungan Jarak Dari Masing-Masing Data Training Ke Data Testing	47
Tabel 4.3.1 Hasil Data Testing 1	49
Tabel 4.3.1 Hasil Data Uji 1	50
Tabel 4.3.1 Hasil Data Testing 2	51
Tabel 4.3.1 Hasil Data Uji 2	52
Tabel 4.3.1 Hasil Data Testing 3	53
Tabel 4.3.1 Hasil Data Uji 3	54
Tabel 4.3.1 Hasil Data Testing 4	55
Tabel 4.3.1 Hasil Data Uji 4	56
Tabel 4.3.1 Hasil Data Testing 5	57
Tabel 4.3.1 Hasil Data Uji 5	58

DAFTAR LAMPIRAN

KODE PROGRAM.....	
DAFTAR RIWAYAT HIDUP.....	
DATASET (<i>TRAINING & TESTING</i>).....	

BAB I

PENDAHULUAN

1. 1 Latar Belakang

Di Indonesia daging ayam merupakan salah satu bahan makanan utama mayoritas masyarakat, Khususnya masyarakat Gorontalo. Karena harga daging ayam dapat dijangkau oleh masyarakat luas. Daging ayam mengandung protein yang tinggi serta berlemak rendah. Murtidjo (2003) memaparkan bahwa daging ayam juga memiliki tekstur yang lebih halus dan lebih lunak jika dibandingkan dengan daging sapi dan ternak lain sehingga lebih mudah dicerna. Namun, sebelum mendapatkan mutu daging ayam yang baik dan layak untuk dimakan oleh masyarakat, [1].

Meningkatnya pemotongan ayam ras pedaging tergantung banyaknya pembeli ayam di kota Gorontalo, pada bulan desember 2018 naiknya jumlah pemotongan ayam di kota Gorontalo hingga 113.750 ekor, menyebabkan para pedagang bersaing untuk memperoleh keuntungan [15]. Hal tersebut memunculkan adanya beberapa pedagang yang tidak bertanggung jawab dalam melakukan perdagangan. Mereka sengaja menjual daging ayam yang sudah tidak layak dikonsumsi dengan tujuan untuk mendapatkan hasil yang sama seperti saat menjual daging normal [19]. Sehingga dalam memilih daging ayam yang baik untuk dikonsumsi harus dilihat dari kesegarannya dan kualitas dagingnya. Namun sebagian masyarakat tidak mengetahui cara membedakan daging ayam segar dan tidak segar.

Menentukan kualitas dan kesegaran daging hasil peternakan ayam broiler dapat dilakukan dengan memperhatikan aroma, warna daging, dan tekstur dagingnya.

Dari sisi aroma, daging hasil peternakan ayam broiler yang segar dan baik serta layak untuk dikonsumsi adalah daging yang tidak berbau amis, berbau menyengat dan asam atau bahkan berbau bahan kimia. Aroma bahan kimia ini cenderung seperti bau obat yang kemungkinan besar daging ayam tersebut telah disuntikkan bahan kimia agar terlihat segar dan layak untuk dikonsumsi. Daging yang berbau bahan kimia ini sebaiknya tidak dikonsumsi karena dapat membahayakan kesehatan tubuh. Selain dari sisi aroma, daging ayam broiler yang sehat, segar, dan berkualitas akan memiliki warna daging yang cerah dan putih kemerahan. Apabila menemukan daging ayam dengan warna yang kemerahan, merah kehitaman atau cenderung biru kehitaman, dapat dipastikan bahwa daging ayam broiler tersebut sudah tidak segar dan meragukan dalam sisi kualitas. Selain melihat dari warna dagingnya, bisa dilihat pula dari serabut daging ayam tersebut. Ayam yang segar dan berkualitas serta layak untuk dikonsumsi adalah ayam broiler yang memiliki serabut daging yang relatif halus. Sedangkan untuk tekstur daging, dilihat dari permukaan daging yang teksturnya lembab dan bila dicubit, kulit daging akan kembali ke bentuk semula, dianjurkan menggunakan sarung tangan atau menggunakan pinset pada saat mencubit kulit ayam. Permukaan daging ayam tidak kering, tidak ada darah dan juga tidak basah. [18].

Beredarnya daging ayam yang tidak segar telah meresahkan dan merugikan masyarakat karena tidak layak dikonsumsi, daging ayam yang tidak segar memiliki kandungan gizi dan secara kasat mata susah untuk dibedakan sehingga dapat menipu masyarakat. Selain kandungan gizi nya berkurang, juga sangat berbahaya untuk kesehatan orang yang mengkonsumsi daging yang sudah tidak segar. Salah satu penyakit yang di timbulkan yaitu demam tinggi, sakit perut, dan sakit kepala [2].

Pengolahan Citra merupakan Metode untuk melakukan klasifikasi terhadap Deteksi kesegaran daging ayam broiler menggunakan tekstur warna, yang mempelajari proses pengolahan gambar dimana baik masukan maupun keluarannya berbentuk berkas citra digital. Juga dijelaskan bahwa pengolahan citra merupakan analisis citra yang banyak melibatkan persepsi visual, maka dari itu dengan judul Deteksi kesegaran daging ayam broiler menggunakan tekstur warna sangat cocok digunakan di Computer Vison, selain mudah dipelajari dan mudah diaplikasikan [3].

KNN (*K-Nearest Neighbor*) merupakan Metode untuk melakukan klasifikasi terhadap Deteksi kesegaran daging ayam broiler menggunakan tekstur warna, yang jaraknya paling dekat atau memiliki persamaan ciri paling banyak dengan objek tersebut. Dekat atau jauhnya tetangga biasanya dihitung dengan jarak *Euclidean*. Teknik ini sederhana dan dapat memberikan akurasi yang baik terhadap hasil klasifikasi (Prasetyo,2012) [16].

Hasil penelitian dengan judul Identifikasi Citra Daging Ayam Berformalin Menggunakan Metode Fitur Tekstur Dan K-Nearest Neighbor (K-NN) menunjukkan bahwa akurasi rata-rata KNN dalam mengidentifikasi daging ayam berformalin dan tidak berformalin sebesar 86,67% [4].

Penggunaan K-NN dan metode GLCM dapat di uji coba, karena algoritma K-NN sangat cepat dalam pelatihan, sederhana dan mudah dipelajari, tahan terhadap data pelatihan yang memiliki derau, dan efektif jika data pelatihan besar [5], dan metode GLCM hanya menggunakan dua inputan dalam fitur ekstaksi yaitu direction dan radius [6], dan menggunakan parameter – parameter yaitu *Contrast*, *Correlation*, *Energy*, *Homogeneity*, dan *Entropy* untuk memperoleh nilai dari fitur. Dengan demikian, algoritma K-NN (K-Nearest Neighbor) dan metode GLCM (Gray Level Co-occurrence Matrix) dapat diuji coba pada identifikasi Deteksi kesegaran daging ayam broiler menggunakan tekstur warna.

Berdasarkan berbagai penerapan di atas, maka peneliti tertarik untuk melakukan penelitian dengan judul Deteksi Kesegaran Daging Ayam *Broiler* Menggunakan Tekstur Warna Dengan Metode *Gray Level CoOccurrence Matrix* Dan *K-Nearest Neighbor*. Diharapkan penelitian ini dapat memberikan kontribusi, berupa penggunaan metode GLCM dengan algoritma K-NN.

1. 2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan maka identifikasi masalah adalah :

1. Masih kurangnya pengetahuan masyarakat tentang cara baik dan tepat memilih ayam segar dan tidak segar yang akan menimbulkan penyakit sehingga akan berdampak buruk bagi kesehatan.
2. Kerugian yang ditimbulkan untuk penjualan ayam tidak segar kepada masyarakat.

1. 3 Rumusan Masalah

Berdasarkan identifikasi masalah maka rumusan masalah penelitian adalah bagaimana hasil Akurasi metode GLCM dan KNN untuk identifikasi kesegaran daging ayam *Broiler*.

1. 4 Tujuan Penelitian

Untuk mengetahui Akurasi metode GLCM dan KNN untuk identifikasi kesegaran daging ayam *Broiler*.

1. 5 Manfaat Penelitian

1.5.1 Manfaat Teoritis

Untuk secara teoritis, diharapkan nanti dengan banyak teori yang didapat dalam studi ini akan lebih menambah wawasan tentang cara mendeteksi kesegaran daging ayam broiler menggunakan tekstur warna dengan metode gray level co-occurrence matrix dan k-nearest neighbor yang belum banyak di ketahui oleh banyak orang.

1.5.2 Manfaat Praktis

Bermanfaat untuk orang banyak nanti untuk lebih belajar mengetahui lebih dalam tentang cara mendeteksi kesegaran daging

ayam broiler menggunakan tekstur warna dengan metode GLCM dan K-NN. Sehingga tidak hanya merugikan keuangan tetapi juga berdampak buruk bagi kesehatan.

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Tabel 2.1 Penelitian Terkait

No	Peneliti	Judul	Tahun	Metode	Hasil
1.	Rosa Andrie Asmara ¹ , Dwi Puspitasari ² , Siti Romlah ³ , Qonitatul H ⁴ , Robertus Romario ⁵ [7].	Identifikasi Kesegaran Daging Sapi Berdasarkan Citranya Dengan Ekstraksi Fitur Warna Dan Teksturnya Menggunakan Metode Gray Level Cooccurrence Matrix	2017	Metode Gray Level Cooccurrence Matrix	Dari hasil perbandingan klasifikasi kesegaran daging sapi menggunakan metode Naïve Bayes, metode Decision Tree J48 dan jaringan syaraf tiruan metode Backpropagation didapatkan bahwa akurasi tertinggi menggunakan metode Naïve Bayes dengan parameter warna dan tekstur atau dengan parameter warna, yaitu dengan nilai akurasi seluruh pengujian sebesar 95.83%
2.	Faris Mushlihul Amin [8].	Identifikasi Citra Daging Ayam Berformalin Menggunakan Metode Fitur Tekstur Dan K-Nearest Neighbor (K-Nn)	2018	Metode Fitur Tekstur Dan K-Nearest Neighbor (K-Nn)	Hasil penelitian menunjukkan bahwa akurasi rata-rata KNN dalam mengidentifikasi daging ayam berformalin dan tidak berformalin sebesar 86,67%.

3.	Faizun Iqbal Zulfi [9].	Identifikasi Tingkat Kesegaran Daging Sapi Lokal Menggunakan Ekstraksi Fitur Warna Berbasis Gui Matlab	2017	Menggunakan Ekstraksi Fitur Warna Berbasis Gui Matlab	Hasil penelitian menunjukkan bahwa Sistem aplikasi pendeteksi tingkat kesegaran daging sapi lokal dapat mengidentifikasi tingkat kesegaran daging sapi segar dengan tingkat keberhasilan sebesar 72,5% dan tingkat kesegaran daging sapi tidak segar dengan tingkat keberhasilan sebesar 80%
----	-------------------------	--	------	---	--

2.2 Tinjauan Pustaka

2.2.1 Analisis Tekstur Warna

2.2.2.1 Ekstraksi Ciri Warna

Ciri warna yang diambil adalah nilai R, G, dan B dari citra sampel. Langkah pertama yang harus dilakukan adalah membuat histogram untuk masing-masing komponen warna yaitu histogram untuk komponen R, komponen G, dan komponen B. Berdasar histogram warna yang sudah dibuat dicari nilai piksel R, G, dan B yang mempunyai tingkat kemunculan paling tinggi. Nilai piksel tersebut yang akan menjadi fitur warna dari sampel citra yang diuji.

2.2.2.2 Ekstraksi Ciri Tekstur

Ciri tekstur yang diambil adalah nilai dimensi fractal citra tersebut. Ciri dimensi fractal dapat digunakan untuk mengetahui tingkat kekasaran tekstur sebuah objek. Objek yang memiliki nilai dimensi fractal yang tinggi berarti objek tersebut memiliki tekstur yang lebih kasar. Langkah pertama yang dilakukan adalah mengubah sampel citra dari ruang warna RGB menjadi citra grayscale. Kemudian dilakukan perhitungan nilai dimensi fraktalnya untuk keseluruhan citra menggunakan algoritma box counting [10].

2.2.2.3 Ekstraksi Ciri Warna

Pada proses ekstraksi ciri warna diawali dengan proses pengubahan aras gambar menjadi gambar beraras keabuan (grayscale). Nilai grayscale suatu piksel akan dimasukkan ke dalam salah satu 8 bin warna. Color histogram menghasilkan 8 fitur keanggotaan. Untuk meminimalkan proses komputasi, masing-masing anggota bin dilakukan normalisasi.

2.2.2.4 Ekstraksi Ciri Tekstur

Ciri tekstur merupakan ciri penting dalam sebuah gambar yang merupakan informasi berupa susunan struktur permukaan suatu gambar. Dalam penelitian ini menggunakan Gray Level oCcurance Matrix (GLCM) sebagai matrik pengambilan nilai keabuan dari sebuah gambar [11].

2.2.2 Computer Vision

Computer vision adalah sebuah kemampuan sebuah computer yang ke desain agar mampu melihat sebuah object sehingga mampu menampilkan objek digital dan bisa mengoleksi data secara visual komputer bisa melakukan beberapa pekerjaan yang tidak bisa dilakukan oleh manusia:

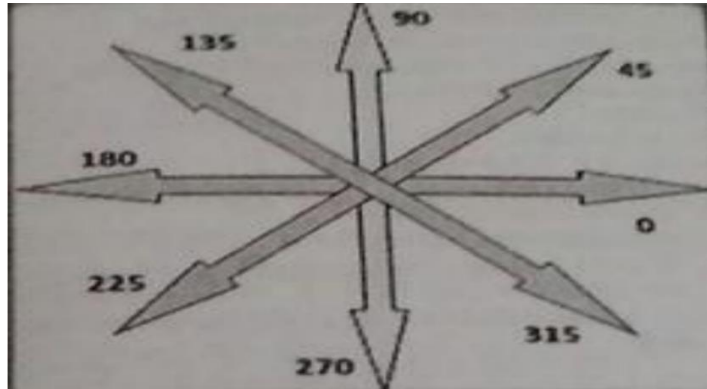
1. Komputer mampu melihat data dalam bentuk pixel bahkan dalam warna yang berbeda.
2. Komputer mampu membandingkan dua object gambar yang sama persis.
3. Komputer mampu melihat sebuah object data selama berjam-jam bahkan sehari – hari. Vision itu sendiri adalah suatu proses evaluasi sebuah data yang bersumber dari image umumnya camera, dengan teknik ekstrasi menggunakan algoritma tertentu.

Dalam istilah sederhana, computer vision adalah bagaimana komputer/mesin dapat melihat, teknik computer vision mampu memvisualisasikan data menganalisa berupa gambar image atau dalam bentuk video. Tujuan utama dari computer vision adalah agar komputer atau mesin dapat meniru kemampuan perseptual mata manusia dan otak, atau bahkan dapat mengunggulinya untuk tujuan tertentu [12].

2.2.3 GLCM (Gray Level Co-Occurrence Matrix)

Matriks ko-okurensi adalah Salah satu metode statistik yang dapat digunakan untuk analisis tekstur. Matrik kookurensi dibentuk dari suatu citra dengan melihat pada piksel-piksel yang berpasangan yang memiliki intensitas tertentu. (Widyaningsih, 2016). Matriks ko-okurensi itu matriks yang

menyatakan hubungan spasial antara dua piksel yang bertetangga yang memiliki intensitas i dan j , yang memiliki jarak d diantara keduanya, dan sudut θ diantara keduanya. Sudut yang digunakan untuk menghasilkan nilai GLCM antara lain Seperti yang ada di Gambar 2.2.3.



Gambar 2.2.3 Delapan arah sudut untuk GLCM

Ekstraksi Ciri adalah proses mengambil ciri-ciri yang terdapat pada objek di dalam citra. Pada proses ini objek di dalam citra dihitung propertiproperti objek yang berkaitan sebagai ciri (Widyaningsih, 2016) [4]. Ciri-ciri tersebut adalah sebagai berikut :

- a. Contrast (Kontras) : Menunjukkan ukuran penyebaran elemen-elemen matriks citra. Jika letaknya jauh dari diagonal utama, nilai kontrasan besar

$$\sum_{i,j=1}^N P(i,j) (i - j)^2$$

- b. Homogeneity (homogenitas) :

$$\sum_{i,j=1}^N \frac{P(i,j)}{1+|i-j|}$$

c. Entropy

$$-\sum_{i,j=0}^{n-1} P(i,j) \log(P(i,j))$$

d. Energy (energi) : Distribusi intensitas piksel terhadap jangkauan aras keabuan.

$$\sum_{i,j=0}^{N-1} P(i-j)^2$$

Dimana, i,j : koordinat di dalam matrik cooccurrence. $P(i,j)$: Matrik Co-occurrence nilai pada koordinat i,j . Dimensi N pada matrik Co-occurrence

e. Dissimilarity

$$\sum_{i,j=1}^N P(i,j)(i-j)$$

Co-Occurrence Matrices adalah matriks yang dibangun menggunakan histogram tingkat kedua. Co-Occurrence Matrices merupakan matriks berukuran $L \times L$ (L menyatakan banyaknya tingkat keabuan) dengan elemen-elemen $P(x_1, x_2)$ yang merupakan distribusi probabilitas bersama (joint probability distribution) dari pasangan pixel dengan tingkat keabuan x_1 yang beralokasi pada koordinat (j,k) dengan x_2 yang beralokasi pada koordinat (m,n) . Koordinat pasangan pixel tersebut berjarak r dengan sudut θ .

Langkah-langkah untuk membuat Co-Occurrence Matrices (matriks co-occurrence) adalah :

- 1) Membuat area kerja matriks.
- 2) Menentukan hubungan spasial piksel referensi dengan piksel tetangga.
- 3) Menghitung jumlah kookurensi dan mengisikannya pada area kerja. 4)
Menjumlahkan matriks kookurensi dengan transposenya untuk
menjadikan-nya simetris
- 5) Normalisasi matriks

Setelah matriks intensitascooccurrence terbentuk, maka tiap elemen matriks $p(i_1, i_2)$ perlu dinormalisasi dengan membagi tiap elemen dengan bilangan yang merupakan jumlah total dari pasangan piksel [13]. Pengukuran nilai tekstur didasarkan pada persamaan Harralick yang didefenisi-kan sebagai berikut:

1. Kontras (Contrast) Menunjukkan ukuran penyebaran (MomentInertia) elemen-elemen matrik citra.

$$Con = \sum_i \sum_j (i - j)^2 P_d(i, j)$$

2. Korelasi (Correlation) Menunjukkan ukuran ketergantungan linear derajat keabuan citra sehingga dapat memberikan petunjuk adanya struktur linear dalam citra

$$Cor = \frac{\sum_i \sum_j (i, j) \cdot P_d(i, j) - \mu_x \cdot \mu_y}{\sigma_x \cdot \sigma_y}$$

3. Momen Selisih Terbalik (Inverse Difference) Disebut juga homogenitas.

Menunjukkan kehomogenan citra.

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

4. Momen Angular Kedua (Angular Second Moment) atau Energy

Menunjukkan ukuran sifat homogenitas citra. Dengan persamaan berikut:

$$ASM = \sum \sum P_d(i, j)^2 \text{ Dimana:}$$

$P_d(i, j)$: Co-Occurrence Matrices ternormalisasi

μ_x : nilai rata-rata elemen kolom pada matriks $P_d(i, j)$

μ_y : nilai rata-rata elemen baris pada matriks $P_d(i, j)$

σ_x : nilai standart deviasi elemen kolom pada matriks

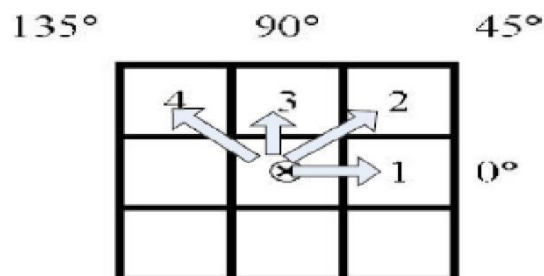
σ_y : nilai standart deviasi elemen baris pada matriks

GLCM adalah sebuah matriks yang menyimpan nilai frekuensi kombinasi perbedaan tingkat kecerahan antara satu piksel dengan piksel di sekitarnya yang terjadi pada sebuah citra Purnomo & Puspitodjati.(2009);. GLCM akan menyimpan informasi mengenai jejak intensitas keabuan antara dua buah piksel yang dipisahkan dengan jarak d dan arah q.

Dalam GLCM arah q dinyatakan dengan sudut Θ_0 (0o, 45o, 90o, 135o).

Sedangkan jarak d menyatakan seberapa jauh jarak antara dua buah piksel yang akan ditinjau Sahaduta & Lubis.(2013);.

Untuk lebih jelasnya, mengenai arah dan jarak dapat dilihat pada Gambar 2.2.3



Gambar 2.2.3 Arah dan Jarak dalam GLCM

Pada Gambar 2.2.3, piksel yang berada di tengah dengan simbol (x) merupakan acuan. Piksel 1, 2, 3, dan 4 memiliki jarak yang sama dari piksel (x) yaitu $d = 1$. Piksel 1 memiliki arah $q = 0^\circ$, piksel 2 memiliki arah $q = 45^\circ$, piksel 3 memiliki arah $q = 90^\circ$, dan piksel 4 memiliki arah $q = 135^\circ$. Untuk menghitung GLCM, terlebih dahulu harus menentukan jarak dan arah yang ingin ditinjau. Langkah selanjutnya adalah membentuk matriks framework yang akan digunakan. Matriks framework ini merepresentasikan range (jangkauan) intensitas derajat keabuan yang dimiliki citra tersebut. Untuk selanjutnya, matriks framework inilah yang akan diisi dengan informasi ketetanggaan (coocurance) dari piksel-piksel yang ada di dalam citra.

Setelah matriks ketetanggaan terisi lengkap, nilai-nilai setiap elemen dalam matriks tersebut harus dinormalisasi. Caranya adalah dengan membagi setiap elemen dalam matriks dengan jumlah total setiap elemen yang ada. Dengan melakukan normalisasi setiap elemen dalam matriks ketetanggaan

dapat diperlakukan sebagai fungsi probabilitas sebab jumlah seluruh elemen akan sama dengan satu. Tujuannya adalah untuk menyeragamkan fitur-fitur GLCM ke dalam cakupan yang sama.

GLCM memiliki 14 fitur yang diajukan oleh Haralick Wibawanto.(2008);, tetapi tidak semua fitur digunakan. Umumnya, hanya 5 fitur saja yang dipakai dalam berbagai penelitian yaitu energi, entropi, kontras, korelasi, dan homogenitas. Hal ini dikarenakan kelima fitur ini merupakan fitur utama dalam GLCM. Sisanya adalah turunan dari 5 fitur ini. Lima fitur GLCM yang digunakan dalam penelitian ini adalah sebagai berikut

Liu.(2006):

1. Entropi

Nilai entropi menunjukkan keteracakan distribusi derajat keabuan suatu citra. Semakin acak distribusi derajat keabuannya, semakin tinggi nilai entropi yang dihasilkan. Entropi didefinisikan seperti persamaan (1).

$$Entropi = -\sum_i \sum_j p(i,j) \log p(i,j) \quad (1)$$

dengan: i = baris j = kolom p = pasangan

matriks intensitas co-occurrence

2. Energi Nilai energi bertolak belakang dengan entropi. Semakin tinggi nilai entropi maka nilai energi akan semakin rendah. Hal ini dikarenakan, nilai energi menggambarkan keteraturan penyebaran derajat keabuan suatu citra. Energi didefinisikan seperti persamaan (2).

$$Energi = \sum_i \sum_j p^2(i,j) \quad (2)$$

3. Kontras

Fitur kontras digunakan untuk menghitung range perbedaan derajat keabuan dalam sebuah citra. Semakin jauh perbedaan derajat keabuan setiap pasangan piksel, semakin tinggi nilai kontras. Demikian sebaliknya, jika perbedaan derajat keabuan setiap pasangan piksel tidak signifikan, nilai kontras akan rendah. Kontras didefinisikan seperti persamaan (3).

$$Kontras = \sum_i \sum_j (i - j)^2 p(i, j) \quad (3)$$

4. Korelasi

Korelasi adalah fitur yang digunakan untuk menghitung ketergantungan linear sebuah citra. Jika derajat keabuan antar pasangan piksel memiliki hubungan linear maka nilai korelasi akan menjadi tinggi. Defini dari korelasi adalah seperti persamaan (4), (5), dan (6).

$$Correlation = \frac{\sum_i \sum_j \frac{(i - \mu)(j - \mu) p(i, j)}{\sigma^2}}{\quad} \quad (4)$$

Dengan:

$$\mu = \sum_i \sum_j i \cdot p(i, j) \quad (5)$$

$$\sigma^2 = \sum_i \sum_j (i - \mu)^2 p(i, j) \quad (6)$$

Keterangan:

μ = nilai rata-rata semua elemen matriks

σ^2 = variansi dari matriks

5. Homogenitas

Fitur homogenitas akan menghitung keseragaman variasi derajat keabuan sebuah citra. Fitur homogenitas akan memiliki nilai yang tinggi derajat keabuan yang hampir sama. Homogenitas didefinisikan seperti persamaan (7) [14].

$$Homogeneity = \sum_i \sum_j \frac{p(i,j)}{1+|i-j|} \quad (7)$$

2.2.4 K-NN (K-Nearest Neighbor)

KNN (*K-Nearest Neighbor*) adalah Metode untuk melakukan klasifikasi terhadap objek berdasar kan data pembelajaran yang jaraknya paling dekat atau memiliki persamaan ciri paling banyak dengan objek tersebut. Dekat atau jauhnya tetangga biasanya dihitung dengan jarak *Euclidean*. Teknik ini sederhana dan dapat memberikan akurasi yang baik terhadap hasil klasifikasi

(Prasetyo,2012).

Rumus:

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \dots + |x_{ip} - x_{jp}|^2}$$

(Persamaan 3)

Keterangan:

$d(i, j)$ = nilai jarak

x_i = nilai – nilai pada fitur 1

x_j = nilai – nilai pada fitur 2

Dalam algoritma ini, nilai k yang terbaik itu tergantung pada jumlah data. Ukuran nilai k yang besar belum tentu menjadi nilai k yang terbaik begitu juga sebaliknya. Tahapan dari klasifikasi menggunakan metode K-NN dalam penelitian ini menurut Prasetyo, (2012) adalah sebagai berikut:

- a. Tahap 1: Penentuan nilai k Penentuan nilai k yang digunakan dalam klasifikasi tidak memiliki aturan yang baku, namun pada penelitian ini nilai k yang digunakan adalah 1,3, 5, 7, 9.
- b. Tahap 2: Perhitungan jarak antar data training dan data uji Teknik perhitungan jarak yang digunakan dalam metode klasifikasi K-NN ini adalah

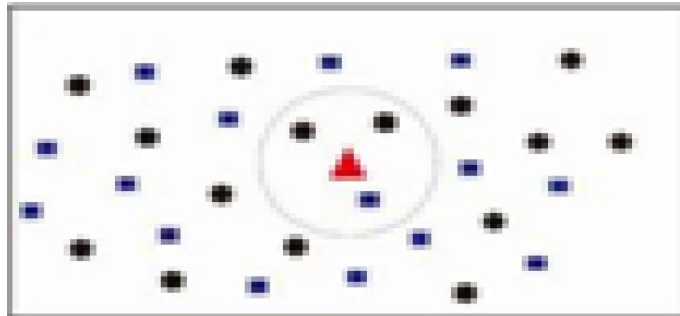
Euclidean

Distance yang direpresentasikan pada rumus 1. Jarak dihitung dari semua data training ke data uji.

- c. Tahap 3: Pengurutan data hasil perhitungan Jarak yang telah didapatkan pada tahap selanjutnya kemudian diurutkan dari yang paling dekat jaraknya sampai yang ke paling jauh (*ascending*).
- d. Tahap 4: Klasifikasi berdasarkan nilai k

Tahap selanjutnya yaitu klasifikasi berdasarkan nilai k . Nilai k atau tetangga terdekat merupakan acuan dari klasifikasi K-NN. Contoh dari penerapan k adalah jika nilai $k = 3$, ini berarti jumlah tetangga terdekat yang dihitung adalah sejumlah 3 tetangga terdekat, dari 3 tetangga terdekat tersebut, diukur mana nilai yang paling mirip dengan objek yang diuji pada pengenalan pola,

contoh penerapan K-NN dapat dilihat pada Gambar 2.2.4



Gambar 2.2.4 Ilustrasi Metode K-NN

Gambar 2.2.4 menjelaskan bahwa terdapat 3 objek, objek pertama yaitu “kotak biru” yang diklasifikasikan sebagai kelas Daun Tempuyung, lalu objek kedua yaitu “bulat hitam” yang diklasifikasikan sebagai kelas bukan Daun Tempuyung, dan objek ketiga yaitu “segitiga merah” yang tidak diketahui termasuk dalam kelas yang mana. Gambar 2.2.4 menampilkan jumlah nilai $k = 3$ dengan merepresentasikan 3 tertangga tertangga terdekat yang berada di dalam lingkaran. Objek yang ada di dalam lingkaran dihitung mana yang paling banyak memiliki kemiripan dengan objek yang sudah diketahui.

Gambar 2 memiliki 2 objek “bulat hitam” dan 1 “kotak biru” di dalam lingkaran tersebut, artinya bisa disimpulkan bahwa objek “segitiga merah” termasuk ke dalam kelas kedua yaitu bukan Daun Tempuyung [15].

K-Nearest Neighbor (KNN) adalah suatu metode pengklasifikasian yang disupervisi, dimana hasil dari query akan di klasifikasikan berdasarkan mayoritas dari kategori. Algoritma ini bertujuan untuk mengklasifikasi objek baru berdasarkan atribut dan training sample. Diberikan suatu titik query, selanjutnya

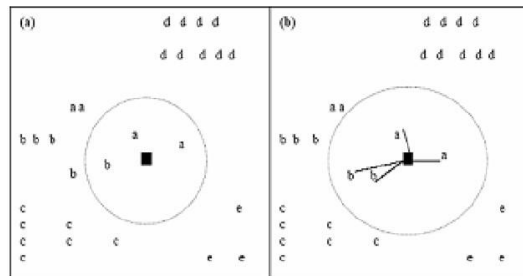
akan ditemukan sejumlah K terdekat ke titik query. Nilai prediksi dari query instance akan ditentukan berdasarkan klasifikasi ketetanggaan [16].

Langkah penyelesaian masalah meng-gunakan metode KNN adalah sebagai berikut:

1. Menentukan nilai parameter K (jumlah tetangga terdekat)
2. Menghitung jarak setiap sample data dengan data yang diuji.
3. Mengurutkan data berdasarkan jarak dari yang terkecil hingga yang terbesar.
4. Mengamati jumlah keputusan yang terbanyak untuk K data yang diambil.
5. Jika terdapat dua atau lebih kelas ω_i yang merupakan tetangga terdekat dari data uji x , maka terjadilah kondisi seimbang (konflik) dan digunakan strategi pemecahan konflik.
6. Untuk masing-masing kelas yang terlibat dalam konflik, tentukan jarak di antara x dengan kelas ω_i berdasarkan E tetangga terdekat yang ditemukan pada kelas ω_i .

7. Jika pola pelatihan ke-m dari kelas ω_i yang terlibat dalam konflik maka jarak antara x dengan kelas ω_i adalah:

$$d_i = \frac{1}{E} \sum_{j=1}^n |(X_j - Y_j)|$$



(a) Model KNN awal (b) model KNN pemecahan konflik

Gambar 2.2.4 Ilustrasi Aturan Pemilihan KNN

Klasifikasi dengan KNN adalah mengelompokkan data baru, dalam penelitian ini adalah data uji berdasarkan jarak ke beberapa data k tetangga terdekat terhadap data latih Hanselman D. (1998); Prinsip kerja KNN adalah menghitung jarak menggunakan jarak Euclidean. Jarak Euclidean digunakan untuk menghitung jarak antara dua vektor yang berfungsi menguji ukuran yang bisa digunakan sebagai interpretasi kedekatan jarak antara dua obyek Mcandrew(2004); yang direpresentasikan dalam persamaan (8).

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (8)$$

Keterangan:

$d(x, y)$: jarak euclidean antara vektor x dan vektor y x_i :

fitur ke i dari vektor x y_i : fitur ke i dari vektor y n :

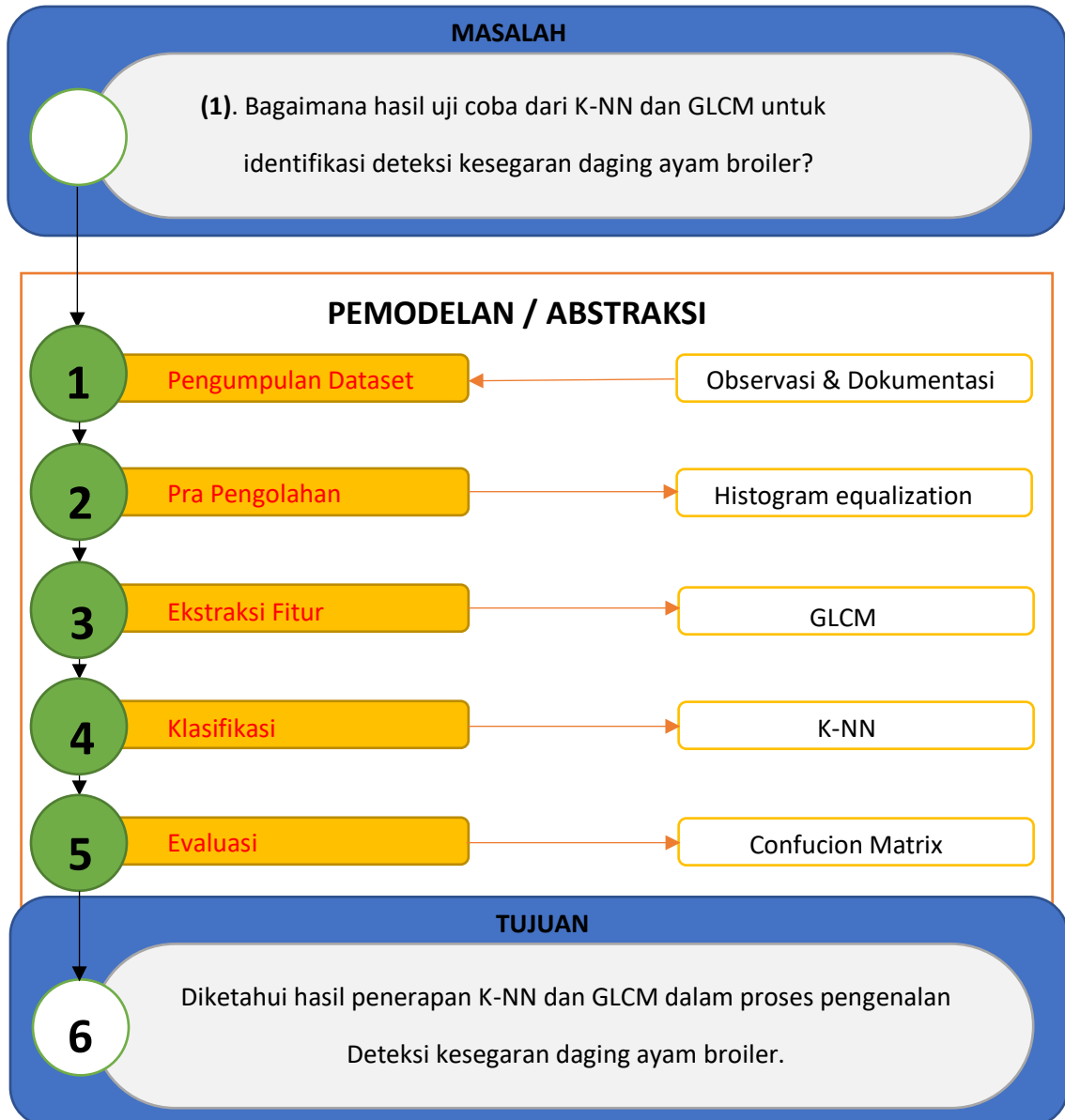
jumlah fitur pada vektor x dan y

Klasifikasi KNN dilakukan dengan mencari k buah tetangga terdekat dari data uji dan memilih kelas dengan anggota terbanyak. Adapun Langkah-langkah klasifikasi KNN adalah sebagai berikut:

1. Mementukan parameter k (jumlah tetangga paling dekat).
2. Menghitung kuadrat jarak Euclidean (query instance) masing-masing objek terhadap data sampel yang diberikan.
3. Kemudian mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak Euclid terkecil.
4. Mengumpulkan kategori y (klasifikasi Nearest Neighbor).
5. Dengan menggunakan kategori Nearest Neighbor yang paling mayoritas maka dapat diprediksi nilai query instance yang telah dihitung.

KNN memiliki beberapa kelebihan yaitu ketangguhan terhadap training data yang memiliki banyak noise dan efektif apabila training data-nya besar. Sedangkan, kelemahan KNN adalah KNN perlu menentukan nilai dari parameter k (jumlah dari tetangga terdekat), training berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil terbaik, dan biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap sampel uji pada keseluruhan sampel latih [17].

2.3 Kerangka Pikir



Gambar 2.3 Pemodelan / Abstraksi

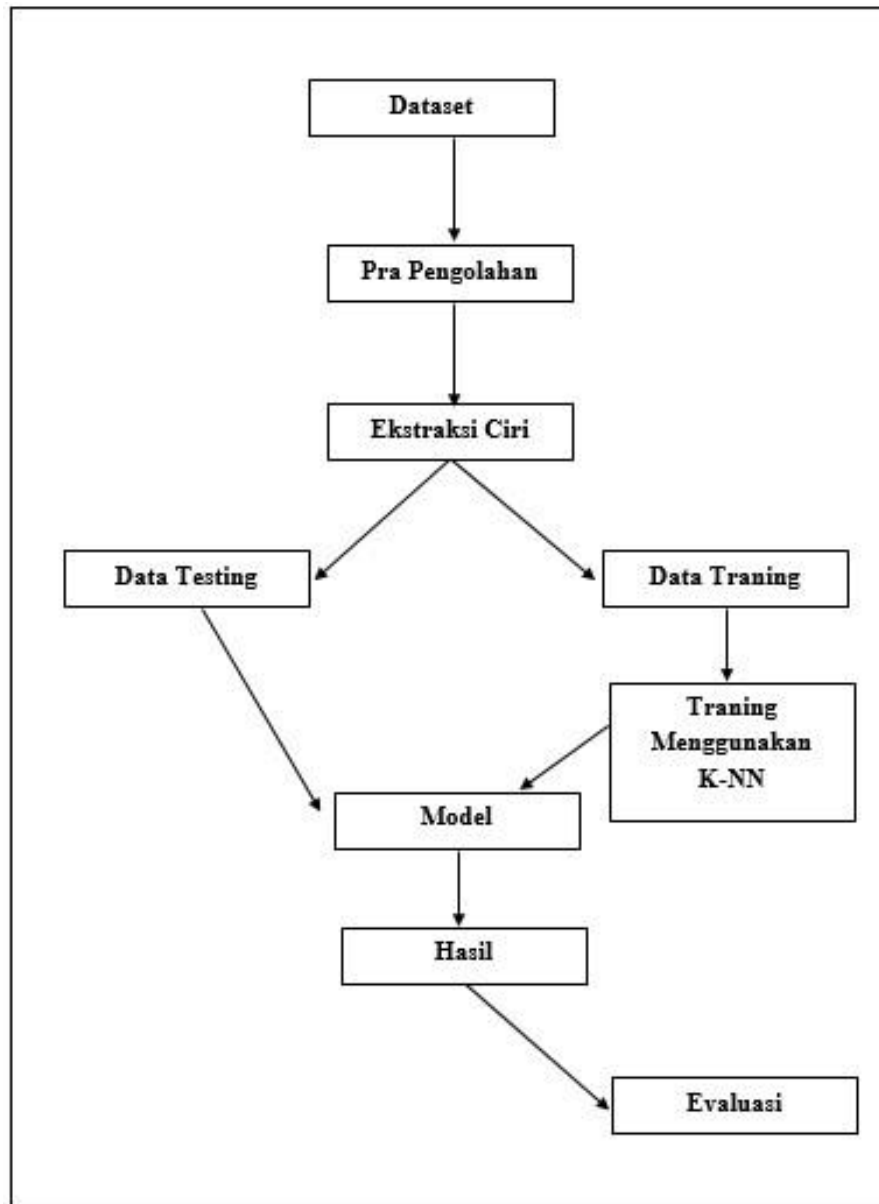
BAB III

METODE PENELITIAN

3. 1. Jenis, Metode, Subjek, Objek, Waktu Dan Lokasi Penelitian

Dipandang dari tingkat penerapannya, maka penelitian ini merupakan penelitian terapan. Dipandang dari jenis informasi yang diolah, maka penelitian ini merupakan penelitian kuantitatif. Dipandang dari perlakuan terhadap data, maka penelitian ini merupakan penelitian konfirmatori. Penelitian ini menggunakan metode penelitian eksperimen. Dengan demikian jenis penelitian ini adalah penelitian eksperimental. Subjek penelitian ini adalah mendeteksi kesegaran daging ayam broiler menggunakan tekstur warna. Penelitian ini dimulai dari September 2019 sampai dengan April 2020

3. 2. Pemodelan



Gambar 3.2 Pemodelan Dataset

3. 2. 1. Pra Pengolahan

Sebelum data diolah, terlebih dahulu dilakukan intensity adjustment dan segmentasi. Hal ini dilakukan karena intensity adjustment bertujuan untuk meningkatkan citra yang gelap dan proses segmentasi ialah proses pemisahan antara objek yang dikehendaki dan objek yang tidak dikehendaki.

3. 2. 2. Ekstraksi Ciri

Ekstraksi ciri berfungsi sebagai deteksi fitur dari suatu citra. Ciri yang dapat digunakan membedakan citra satu dengan citra yang lain, di antaranya adalah ciri tekstur dan warna. Pada tahap ini digunakan Gray Level Co-occurrence Matrix. Masing-masing citra diekstrak cirinya berdasarkan parameter-parameternya tertentu dan dikelompokkan pada kelas tertentu, parameter – parameter yang digunakan berupa Contrasts, Correlation, Energy, Homogeneity, dan Entropy.

3. 2. 3. Data Training

Data Training berupa data yang telah terekstrak cirinya yang selanjutnya akan dilatih menggunakan algoritma K-Nearest Neighbor, jumlah data training yang digunakan berupa 90% dari sampel daging ayam broiler dengan deteksi kesegaran yang berbeda. Algoritma ini akan menentukan/mencari bobot yang terbaik. Data training ini berupa hasil dari ekstraksi Gray Level Co-occurrence Matrix terhadap deteksi kesegaran daging ayam broiler menggunakan tekstur warna.

3. 2. 4. Traning Menggunakan K-NN

Traning menggunakan K-NN yaitu menjadikan data traning menjadi data inputan. Algoritma KNN bekerja berdasarkan jarak terpendek dari training dan test dalam penentuannya, kemudian mengambil mayoritas hasil ketentuan yang telah didapatkan untuk dijadikan prediksi dari test.

3. 2. 5. Model

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

3. 2. 6. Data Testing

Data testing merupakan data yang telah terekstraksi cirinya, untuk menguji data yang telah dilatih, jumlah data traning yang digunakan berupa 90% dari sampel daging ayam broiler dengan deteksi kesegaran yang berbeda. Data testing digunakan untuk mengetahui keberhasilan classifier berhasil melakukan klasifikasi, data testing merupakan ciri dari hasil ekstraksi dari Gray Level Co-occurrence Matrix terhadap deteksi kesegaran daging ayam broiler menggunakan tekstur warna.

3. 2. 7. Hasil Klasifikasi

Hasil klasifikasi merupakan output, pada data testing yang didapatkan dari proses klasifikasi yang menggunakan algoritma K-NN berdasarkan model yang diperoleh dari data traning

3. 2. 8. Evaluasi

Evaluasi bertujuan untuk mengetahui hasil kinerja dari metode yang digunakan, evaluasi dilakukan pada data testing dan output yang dihasilkan akan dimasukkan ke dalam Confusion Matrix untuk menghitung nilai akurasi.

BAB IV

HASIL PENELITIAN

4. 1. Hasil Pengumpulan Data

Dataset yang digunakan pada penelitian ini ada berupa data real, Data terdiri dari 100 citra dan 3 objek (ayam segar, ayam freezer dan ayam busuk), dengan format file gambar adalah jpg. Berikut ini beberapa sampel gambar yang berhasil dilakukan :



Gambar 4. 1. Perbedaan Kesegaran Daging Ayam

Untuk melakukan proses analisis pada penelitian ini dengan melakukan pemilihan data gambar yang dijadikan sebagai data training dan data testing, sehingganya dari pekerjaan yang di lakukan pada penelitian ini telah di tentukan data yang berhasil di dapatkan ditentukan jumlah data training sebanyak 90 data gambar dan data testing sebanyak 10 data gambar.

Tabel 4.1 Mengkoversi RGB ke Grayscale

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

HISTOGRAM EQUALIZATION

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

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

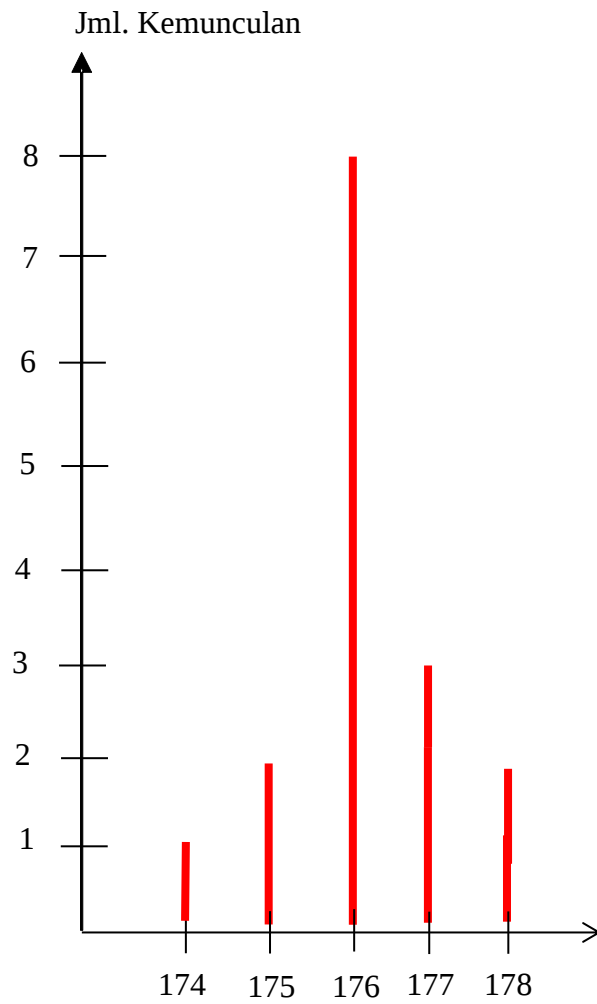
$$= \sum_{j=0}^k \frac{n_j}{n}$$

Sebagai Contoh didapat sebuah citra dengan image 4x4

Tabel 4.1 Citra Dengan Image 4x4

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

Dengan Histogram sebagai berikut :



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

Tabel 4.1 Perhitungan Dengan Langkah Dan Rumus

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

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

Tabel 4.1 Citra 4x4 Asli

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



Tabel 4.1 Data Citra 4x4 Setelah Dilakukan Histogram Equalisation

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

Segmentasi Citra

Segmentasi citra merupakan bagian dari proses pengolahan citra. Segmentasi citra (*image segmentation*) mempunyai arti membagi suatu citra menjadi wilayah-wilayah yang homogen berdasarkan kriteria keserupaan tertentu antara tingkat keabuan suatu piksel dengan tingkat keabuan piksel-piksel tetangganya, kemudian hasil dari proses segmentasi ini akan digunakan untuk proses lebih lanjut [20].

Metode Otsu *Thresholding*

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

Untuk mendapatkan nilai *Threshold* ada perhitungan yang harus dilakukan. Langkah awal yang harus dilakukan adalah membuat histogram. Dari histogram dapat diketahui jumlah piksel untuk setiap tingkat keabuan. Tingkat keabuan citra dinyatakan dengan i sampai dengan L . Level ke i dimulai dari 1, yaitu piksel 0. Untuk L , maksimal level adalah 256 dengan piksel bernilai 255.

Nilai ambang yang akan dicari dari suatu citra *grayscale* dinyatakan dengan k . Nilai k berkisar antara 0 sampai dengan $L-1$ dengan nilai $L=256$ (simbol histogram adalah P_i). Jadi probabilitas setiap piksel pada level ke i dinyatakan dengan persamaan (1):

$$P_i = \frac{n_i}{N} \quad (2)$$

Keterangan:

P_i = Probabilitas piksel ke- i n_i = Jumlah piksel dengan tingkat keabuan i

N = Total jumlah piksel pada citra

Langkah selanjutnya mencari nilai jumlah kumulatif, rerata kumulatif dan intensitas global. mencari nilai tersebut dapat melihat persamaan (3), persamaan (4), dan persamaan (5).

Berikut adalah formulasi untuk menghitung jumlah kumulatif (*cumulative sum*) dari $\square(k)$, untuk

$L = 0, 1, 2, \dots, L-1$:

$$\omega(k) = \sum_{i=0}^k p_i \quad (3)$$

Berikut adalah formulasi untuk menghitung rerata kumulatif (*cumulative mean*) dari $\square(k)$, untuk

$L = 0, 1, 2, \dots, L-1$:

$$\mu(k) = \sum_{i=0}^k i \cdot p_i \quad (4)$$

Berikut adalah formulasi untuk menghitung rerata intensitas global $\square_T \square k \square$:

$$\mu_T(k) = \sum_{i=0}^{L-1} i \cdot p_i \quad (5)$$

Pada persamaan (3), persamaan (4), maupun persamaan (5), nilai k menyatakan tingkat level keabuan dimana setiap rentang piksel akan dihitung. Langkah selanjutnya adalah menentukan varian antar kelas (*between class variance*).

Persamaan untuk *between class variance* (6):

$$\sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \quad (6)$$

Hasil dari perhitungan *between class variance* dicari nilai maksimal. Nilai yang paling besar digunakan sebagai *threshold* atau nilai ambang (k), dengan persamaan (7):

$$\sigma_B^2(k^*) = \max_{1 \leq x \leq L} \sigma_B^2(k) \quad (7)$$

Keterangan:

$\omega(k)$ = Jumlah Kumulatif
 $\mu(k)$
 $\mu_T(k)$ = Rerata Kumulatif
 σ_B^2 = Rerata Intensitas Global
 = Nilai Ambang

Between class variance bertujuan untuk mencari nilai ambang dari sebuah citra *grayscale*, nilai ambang atau *threshold* digunakan sebagai nilai acuan untuk mengubah citra *grayscale* ke citra biner. Setiap citra memiliki nilai ambang yang berbeda-beda.

4. 2. Hasil Pemodelan

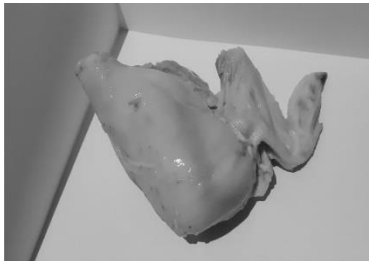
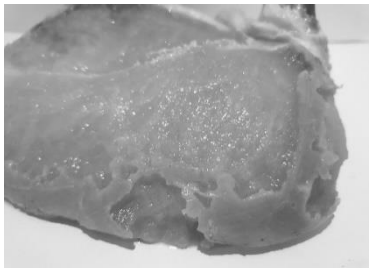
4. 2. 1. Pra Pengolahan

Pra-pengolahan citra (*image pre-processing*), yaitu proses paling awal dalam pengolahan citra sebelum proses utama dilakukan. Pada tahap ini citra daging ayam *broiler* yang sudah ada dikonversi agar diperoleh data citra daging ayam *broiler* yang sesuai kebutuhan. Tahap ini berfungsi untuk menormalisasi citra daging ayam broiler dari permasalahan luminasi yang terlalu gelap atau terlalu terang sehingga dapat meningkatkan performansi dari sistem pengenalan daging ayam *broiler*. Pra-pengolahan dalam penelitian ini dilakukan dengan 2 tahap, yaitu:

1. Pengubahan citra warna ke *grayscale*

Pra-pengolahan pertama yang akan dilakukan adalah merubah dataset yang awalnya berbentuk citra dari RGB (*red, green, blue*) menjadi citra bentuk *grayscale*, perubahan ini dilakukan karena citra *grayscale* memiliki persamaan yang sederhana dan mampu mengurangi kebutuhan *memory* dimana nilai warna putih diwakili dengan angka 255 dan nilai warna hitam diwakili dengan angka 0. Berikut gambar proses perubahan citra warna ke *grayscale*

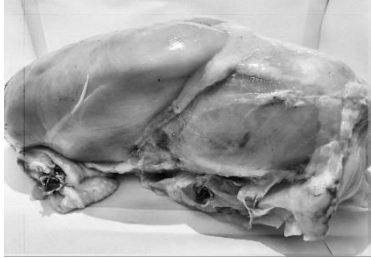
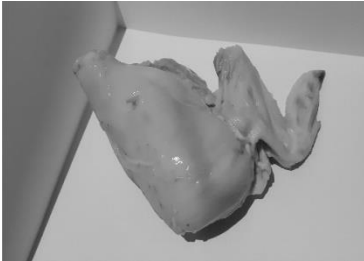
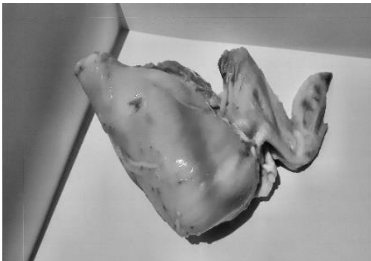
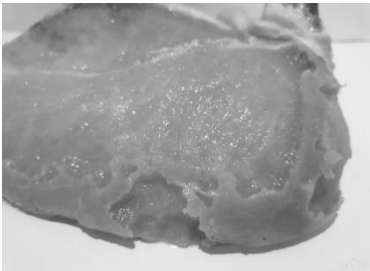
Tabel 4.2.1 Proses Perubahan Citra Asli Menjadi *Grayscale*

Perbedaan	<i>Images Original</i>	<i>Images Grayscale</i>
Ayam Segar		
Ayam Freezer		
Ayam Busuk		

2. Normalisasi citra

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

Tabel 4.2.1 Proses Normalisasi Citra

Perbedaan	Citra <i>Grayscale</i>	Citra <i>Histogram</i>
Ayam Segar		
Ayam Freezer		
Ayam Busuk		

4. 2. 2. Ekstraksi Fitur

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

Contoh: Jarak 1 dan orientasi arah sudut 0°

Nilai dari Tabel 4.2.2 berupa *matrix* 10 x 10 yang sebenarnya *matrix* 255 x 255 yang disederhanakan untuk memudahkan perhitungan.

Tabel 4.2.2 Nilai Dari Suatu Citra

	0	1	2	3	4	5	6	7	8	9
0	176	176	176	175	176	176	178	179	177	179
1	177	177	176	177	177	180	178	177	179	180
2	175	176	176	178	177	178	178	178	178	179
3	176	176	174	178	176	178	177	179	178	177
4	175	177	175	179	178	178	177	177	177	179
5	175	180	176	179	177	178	178	176	179	179
6	175	175	177	178	178	178	179	178	176	179
7	176	179	177	177	178	178	177	180	179	178
8	177	176	177	177	177	178	179	180	177	178
9	177	177	178	177	178	177	179	178	178	178

Tabel 4.2.2 *Matrix GLCM*

i/j	175	176	177	178	179	180
175	1	2	1	0	0	1
176	1	2	2	2	1	0
177	2	1	2	2	2	1
178	0	1	2	3	1	1
179	0	1	1	2	1	1
180	1	1	1	1	1	0

Jumlah Total nilai *GLCM* = 42

Tabel 4.2.2 *Matrix Ternormalisasi*

i/j	175	176	177	178	179	180
175	0,023	0,047	0,023	0	0	0,023
176	0,023	0,047	0,047	0,047	0,023	0
177	0,047	0,023	0,047	0,047	0,047	0,023
178	0	0,023	0,047	0,071	0,023	0,023
179	0	0,023	0,023	0,047	0,023	0,023
180	0,023	0,023	0,023	0,023	0,023	0

a. Menghitung nilai *Contras*

$$\begin{aligned} \text{Contras} = & (175-175)^2(0,023) + (176-175)^2(0,047) + (177-175)^2(0,023) + (180- \\ & 175)^2(0,023) + (175-176)^2(0,023) + (176-176)^2(0,047) + (177-176)^2(0,047) + (178- \\ & 176)^2(0,047) + (179-176)^2(0,023) + (175-177)^2(0,047) + (176-177)^2(0,023) + (177- \\ & 177)^2(0,047) + (178-177)^2(0,047) + (179-177)^2(0,047) + (180-177)^2(0,023) + (176- \\ & 178)^2(0,023) + (177-178)^2(0,047) + (178-178)^2(0,071) + (179-178)^2(0,023) + (180- \\ & 178)^2(0,023) + (176-179)^2(0,023) + (177-179)^2(0,023) + (178-179)^2(0,047) + (179- \\ & 179)^2(0,023) + (180-179)^2(0,023) + (175-180)^2(0,023) + (176-180)^2(0,023) + (177- \\ & 180)^2(0,023) + (178-180)^2(0,023) + (179-180)^2(0,023) \end{aligned}$$

$$\begin{aligned} \text{Contras} = & 0 + 0,50 + 0,90 + 0,60 + 0,20 + 0 + 0,50 + 0,20 + 0,21 + 0,19 + 0,23 + 0 \\ & + 0,48 + 0,20 + 0,21 + 1,00 + 1,00 + 0 + 0,23 + 1,00 + 0,21 + 1,00 + 0,47 + 0 + \\ & 0,23 + 0,60 + 0,76 + 0,21 + 1,00 + 0,23 \end{aligned}$$

$$\text{Contras} = \mathbf{12,36}$$

b. Menghitung nilai *Homogeneity*

$$\begin{aligned} \text{Homogeneity} = & ((0,023) / 1 + (175-175)) + ((0,047) / 1 + (176-175)) + ((0,023) / 1 + \\ & (177-175)) + ((0,023) / 1 + (180-175)) + ((0,023) / 1 + (175-176)) + ((0,047) / 1 + \\ & (176-176)) + ((0,047) / 1 + (177-176)) + ((0,047) / 1 + (178-176)) + ((0,023) / 1 + \\ & (179-176)) + ((0,047) / 1 + (175-177)) + ((0,023) / 1 + (176-177)) + ((0,047) / 1 + \\ & (177-177)) + ((0,047) / 1 + (178-177)) + ((0,047) / 1 + (179-177)) + ((0,023) / 1 + \\ & (180-177)) + ((0,023) / 1 + (176-178)) + ((0,047) / 1 + (177-178)) + ((0,071) / 1 + \\ & (178-178)) + ((0,023) / 1 + (179-178)) + ((0,023) / 1 + (180-178)) + ((0,023) / 1 + \\ & (176-179)) + ((0,023) / 1 + (177-179)) + ((0,047) / 1 + (178-179)) + ((0,023) / 1 + \\ & (179-179)) + ((0,023) / 1 + (180-179)) + ((0,023) / 1 + (175-180)) + ((0,023) / 1 + \\ & (176-180)) + ((0,023) / 1 + (177-180)) + ((0,023) / 1 + (178-180)) + ((0,023) / 1 + \\ & (179-180)) \end{aligned}$$

$$\text{Homogeneity} = \mathbf{-3,022}$$

c. Menghitung nilai *Entropy*

$$\begin{aligned} Entropy = & (0,023 \cdot \log 0,023) + (0,047 \cdot \log 0,047) + (0,023 \cdot \log 0,023) + (0,023 \cdot \log \\ & 0,023) + (0,023 \cdot \log 0,023) + (0,047 \cdot \log 0,047) + (0,047 \cdot \log 0,047) + (0,047 \cdot \log \\ & 0,047) + (0,023 \cdot \log 0,023) + (0,047 \cdot \log 0,047) + (0,023 \cdot \log 0,023) + (0,047 \cdot \log \\ & 0,047) + (0,047 \cdot \log 0,047) + (0,047 \cdot \log 0,047) + (0,023 \cdot \log 0,023) + (0,023 \cdot \log \\ & 0,023) + (0,047 \cdot \log 0,047) + (0,071 \cdot \log 0,071) + (0,023 \cdot \log 0,023) + (0,023 \cdot \log \\ & 0,023) + (0,023 \cdot \log 0,023) + (0,023 \cdot \log 0,023) + (0,047 \cdot \log 0,047) + (0,023 \cdot \log \\ & 0,023) + (0,023 \cdot \log 0,023) + (0,023 \cdot \log 0,023) + (0,023 \cdot \log 0,023) + (0,023 \cdot \log \\ & 0,023) + (0,023 \cdot \log 0,023) + (0,023 \cdot \log 0,023) \end{aligned}$$

$$Entropy = -1,4215995997$$

d. Menghitung nilai *Energy*

$$\begin{aligned} Energy = & ((0,023)^2) + ((0,047)^2) + ((0,023)^2) + ((0,023)^2) + ((0,023)^2) + ((0,047)^2) \\ & + ((0,047)^2) + ((0,047)^2) + ((0,023)^2) + ((0,047)^2) + ((0,023)^2) + ((0,047)^2) + \\ & ((0,047)^2) + ((0,047)^2) + ((0,023)^2) + ((0,023)^2) + ((0,047)^2) + ((0,071)^2) + \\ & ((0,023)^2) + ((0,023)^2) + ((0,023)^2) + ((0,023)^2) + ((0,047)^2) + ((0,023)^2) + \\ & ((0,023)^2) + ((0,023)^2) + ((0,023)^2) + ((0,023)^2) + ((0,023)^2) + ((0,023)^2) \end{aligned}$$

$$Energy = 0,037182$$

4. 2. 3. Klasifikasi

Tabel 4.2.3 Hasil Perhitungan Data *Training*

No	Data	Con(X1)	Hom(X2)	Ent(X3)	Ene(X4)	Y
1		183.083875	0.62069490	4.85224623	0.37142602	Ayam
		34635682	27834016	7501204	186123896	Busuk
2		227.629361	0.61352745	4.86033754	0.36325627	Ayam
		78151085	61092432	56548585	581857667	Busuk
3		222.692802	0.58020078	5.17895109	0.28378781	Ayam
		98257308	35920091	01874305	545566073	Busuk
4		226.580115	0.57248088	5.27426237	0.29874907	Ayam
		13613428	82219949	9886439	481215823	Busuk
5		295.871415	0.75867761	3.24534072	0.59863436	Ayam
		83941417	00805026	02590134	29282341	Busuk

6		150.720306	0.79398901	2.95191892	0.65020459	Ayam
		86018707	27133541	69436423	11690653	Segar
7		109.775777	0.71846179	4.08580216	0.48534663	Ayam
		71614469	8310835	2433137	778806897	Segar
8		85.0508068	0.71545041	4.04156539	0.49196716	Ayam
		9820198	1042428	6216265	517678896	Segar
						
10		175.369373	0.76652098	3.65767795	0.52614112	Ayam
		33740468	67492531	1575042	66166133	Segar

Tabel 4.2.3 Hasil Perhitungan Data *Testing*

No	Data	Con(X1)	Hom(X2)	Ent(X3)	Ene(X4)	Y
1		121.387651	0.71705994	4.04873401	0.50163868	?
		39160904	65209052	813019	66559918	

Euclidean Distance

$$\begin{aligned}
 d &= \sqrt{(183.0838 - 121.3876)^2 + (0.6206 - 0.7170)^2} \\
 &\quad (4.8522 - 4.0487)^2 + (0.3714 - 0.5016)^2 \\
 &= \sqrt{3.806.4210 + 0.0092} \\
 &\quad 0.6456 + 0.0169 \\
 &= \sqrt{3.807.0927} \\
 &= 61.7016
 \end{aligned}$$

Tabel 4.2.3 Perbandingan Nilai

No	Perhitungan Jarak <i>Euclidean Distance</i>	Nilai
1	$d = \sqrt{3.806.4210 + 0.0092} + 0.6456 + 0.0169$	61.7016
2	$d = \sqrt{11.287.2988 + 0.0107} + 0.6586 + 0.0191$	106.2449
3	$d = \sqrt{10.262.7435 + 0.0187} + 1.2773 + 0.0474$	101.3118
4	$d = \sqrt{11.065.4620 + 0.0209} + 1.5018 + 0.0411$	105.1999
5	$d = \sqrt{30.444.5964 + 0.0017} + 0.6454 + 0.0094$	174.4856

6	$d = \sqrt{860.4072 + 0.0059 + 1.2029 + 0.0220}$	29.3536
7	$d = \sqrt{134.8362 + 0.00000196 + 0.0013 + 0.0002}$	11.6119
8	$d = \sqrt{1.320.3630 + 0.00000256 + 0.00005184 + 0.00009409}$	36.3368
		
10	$d = \sqrt{2.914.0239 + 0.0024 + 0.1529 + 0.0006}$	53.9831

Tabel 4.2.3 Hasil Pengurutan Dari Nilai Terkecil Berdasarkan Perhitungan Jarak Dari Masing-Masing Data *Training* Ke Data *Testing*

No	<i>Euclidean</i>	Y
7	11.6119	Ayam Segar
6	29.3536	Ayam Segar
8	36.3368	Ayam Segar
10	53.9831	Ayam Segar
1	61.7016	Ayam Busuk
3	101.3118	Ayam Busuk
4	105.1999	Ayam Busuk
2	106.2449	Ayam Busuk
5	174.4856	Ayam Busuk

No	<i>Euclidean</i>	Y
7	11.6119	Ayam Segar
6	29.3536	Ayam Segar
8	36.3368	Ayam Segar
10	53.9831	Ayam Segar
1	61.7016	Ayam Busuk
3	101.3118	Ayam Busuk

Jumlah K=6 Yang Telah Ditentukan

Y	Jumlah Data
Ayam Segar	4
Ayam Busuk	2

Tetangga Paling Banyak Muncul = Ayam Segar

4. 3. Evaluasi

4. 3. 1. Confusion Matrix

Pada penelitian ini menggunakan confusion Matrix sebagai metode. Dalam perhitungan akurasi pada penerapan deteksi kesegaran daging ayam *broiler* menggunakan tekstur warna. Evaluasi kinerja deteksi kesegaran daging ayam *broiler* di dasarkan dari jumlah pengujian objek.

Contoh: Jarak 1, orientasi arah sudut 0° dan nilai K 1

Tabel 4.3.1 Hasil Data Testing 1

No.	Image Testing	Data Aktual	Hasil Klasifikasi
1.		Ayam Segar	Ayam Segar
2.		Ayam Segar	Ayam Freezer
3.		Ayam Segar	Ayam Freezer
4.		Ayam Freezer	Ayam Segar
5.		Ayam Freezer	Ayam Freezer

6.		Ayam Freezer	Ayam Freezer
7.		Ayam Busuk	Ayam Busuk
8.		Ayam Busuk	Ayam Busuk
9.		Ayam Busuk	Ayam Busuk
10.		Ayam Busuk	Ayam Segar

Tabel 4.3.1 Hasil Data Uji 1

	Ayam Segar	Ayam Freezer	Ayam Busuk
Ayam Segar	1	2	0
Ayam Freezer	1	2	0
Ayam Busuk	1	0	3

Jumlah data yang diklasifikasi dengan benar = 6

Jumlah data yang diklasifikasi salah = 4

$$\text{Akurasi} = \frac{\text{Data Uji Dikenali}}{\text{Data uji}} \times 100 \% \quad \text{Akurasi} = \frac{6}{10} \times 100 \% = 60 \%$$

Contoh: Jarak 1, orientasi arah sudut 0° dan nilai K 2

Tabel 4.3.1 Hasil Data Testing 2

No.	Image Testing	Data Aktual	Hasil Klasifikasi
1.		Ayam Segar	Ayam Busuk
2.		Ayam Segar	Ayam Freezer
3.		Ayam Segar	Ayam Freezer
4.		Ayam Freezer	Ayam Freezer
5.		Ayam Freezer	Ayam Freezer
6.		Ayam Freezer	Ayam Freezer
7.		Ayam Busuk	Ayam Busuk

8.		Ayam Busuk	Ayam Busuk
9.		Ayam Busuk	Ayam Busuk
10.		Ayam Busuk	Ayam Busuk

Tabel 4.3.1 Hasil Data Uji 2

	Ayam Segar	Ayam Freezer	Ayam Busuk
Ayam Segar	0	2	1
Ayam Freezer	0	3	0
Ayam Busuk	0	0	4

Jumlah data yang diklasifikasi dengan benar = 7

Jumlah data yang diklasifikasi salah = 3

$$\text{Akurasi} = \frac{\text{Data Uji Dikenali}}{\text{Data uji}} \times 100 \% \quad \text{Akurasi} = \frac{7}{10} \times 100 \% = 70 \%$$

Contoh: Jarak 1, orientasi arah sudut 0° dan nilai K 3

Tabel 4.3.1 Hasil Data Testing 3

No.	Image Testing	Data Aktual	Hasil Klasifikasi
1.		Ayam Segar	Ayam Busuk
2.		Ayam Segar	Ayam Segar
3.		Ayam Segar	Ayam Freezer
4.		Ayam Freezer	Ayam Freezer
5.		Ayam Freezer	Ayam Segar
6.		Ayam Freezer	Ayam Segar

7.		Ayam Busuk	Ayam Busuk
8.		Ayam Busuk	Ayam Busuk
9.		Ayam Busuk	Ayam Busuk
10.		Ayam Busuk	Ayam Busuk

Tabel 4.3.1 Hasil Data Uji 3

	Ayam Segar	Ayam Freezer	Ayam Busuk
Ayam Segar	1	1	1
Ayam Freezer	2	1	0
Ayam Busuk	0	0	4

Jumlah data yang diklasifikasi dengan benar = 6

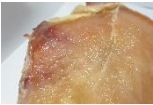
Jumlah data yang diklasifikasi salah = 4

$$\text{Akurasi} = \frac{\text{Data Uji Dikenali}}{\text{Data uji}} \times 100 \% \quad \text{Akurasi} = \frac{6}{10} \times 100 \% = 60 \%$$

Contoh: Jarak 1, orientasi arah sudut 0° dan nilai K 4

Tabel 4.3.1 Hasil Data Testing 4

No.	Image Testing	Data Aktual	Hasil Klasifikasi
1.		Ayam Segar	Ayam Segar
2.		Ayam Segar	Ayam Freezer
3.		Ayam Segar	Ayam Freezer
4.		Ayam Freezer	Ayam Freezer
5.		Ayam Freezer	Ayam Freezer
6.		Ayam Freezer	Ayam Freezer

7.		Ayam Busuk	Ayam Busuk
8.		Ayam Busuk	Ayam Busuk
9.		Ayam Busuk	Ayam Busuk
10.		Ayam Busuk	Ayam Busuk

Tabel 4.3.1 Hasil Data Uji 4

	Ayam Segar	Ayam Freezer	Ayam Busuk
Ayam Segar	1	2	0
Ayam Freezer	0	3	0
Ayam Busuk	0	0	4

Jumlah data yang diklasifikasi dengan benar = 8

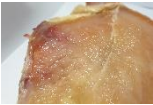
Jumlah data yang diklasifikasi salah = 2

$$\text{Akurasi} = \frac{\text{Data Uji Dikenali}}{\text{Data uji}} \times 100 \% \quad \text{Akurasi} = \frac{8}{10} \times 100 \% = 80 \%$$

Contoh: Jarak 1, orientasi arah sudut 0° dan nilai K 5

Tabel 4.3.1 Hasil Data Testing 5

No.	Image Testing	Data Aktual	Hasil Klasifikasi
1.		Ayam Segar	Ayam Freezer
2.		Ayam Segar	Ayam Segar
3.		Ayam Segar	Ayam Segar
4.		Ayam Freezer	Ayam Freezer
5.		Ayam Freezer	Ayam Freezer
6.		Ayam Freezer	Ayam Freezer

7.		Ayam Busuk	Ayam Busuk
8.		Ayam Busuk	Ayam Busuk
9.		Ayam Busuk	Ayam Busuk
10.		Ayam Busuk	Ayam Busuk

Tabel 4.3.1 Hasil Data Uji 5

	Ayam Segar	Ayam Freezer	Ayam Busuk
Ayam Segar	2	1	0
Ayam Freezer	0	3	0
Ayam Busuk	0	0	4

Jumlah data yang diklasifikasi dengan benar = 9

Jumlah data yang diklasifikasi salah = 1

$$\text{Akurasi} = \frac{\text{Data Uji Dikenali}}{\text{Data uji}} \times 100 \% \quad \text{Akurasi} = \frac{9}{10} \times 100 \% = 90 \%$$

BAB V

PEMBAHASAN

4. 4. Pembahasan Model

Data yang digunakan dalam penelitian ini terdiri dari data real peneliti yang di dapat dengan mengambil gambar secara langsung. Masing-masing 30 data training dan 10 data testing sehingga total dataset yang di gunakan pada penelitian ini yaitu 100 data set. Berdasarkan hasil penelitian data yang di ambil terdiri dari 3 kategori yaitu Segar, Freezer dan Busuk. Selanjutnya citra daging ayam di ekstraksi ciri menggunakan *Gray Level Co-occurrence Matrix*. Setelah di dapatkan ciri dari citra daging ayam, selanjutnya dilakukan proses klasifikasi menggunakan *K-Nearest Neighbor*. Pada tahap ini citra pada daging ayam dijadikan sebagai inputan untuk mencari nilai akurasi terbaik pada percobaan model K-NN.

Pada percobaan model K-NN, dengan jumlah $K = 2$, dan Arah GLCM = 90° , dengan jarak = 3, Maka Mendapatkan Nilai Akurasi Sebesar = 90%, dengan Data Training = 90, yaitu data ayam segar = 30, data ayam freezer = 30, dan data ayam busuk = 30, Data Testing = 10.

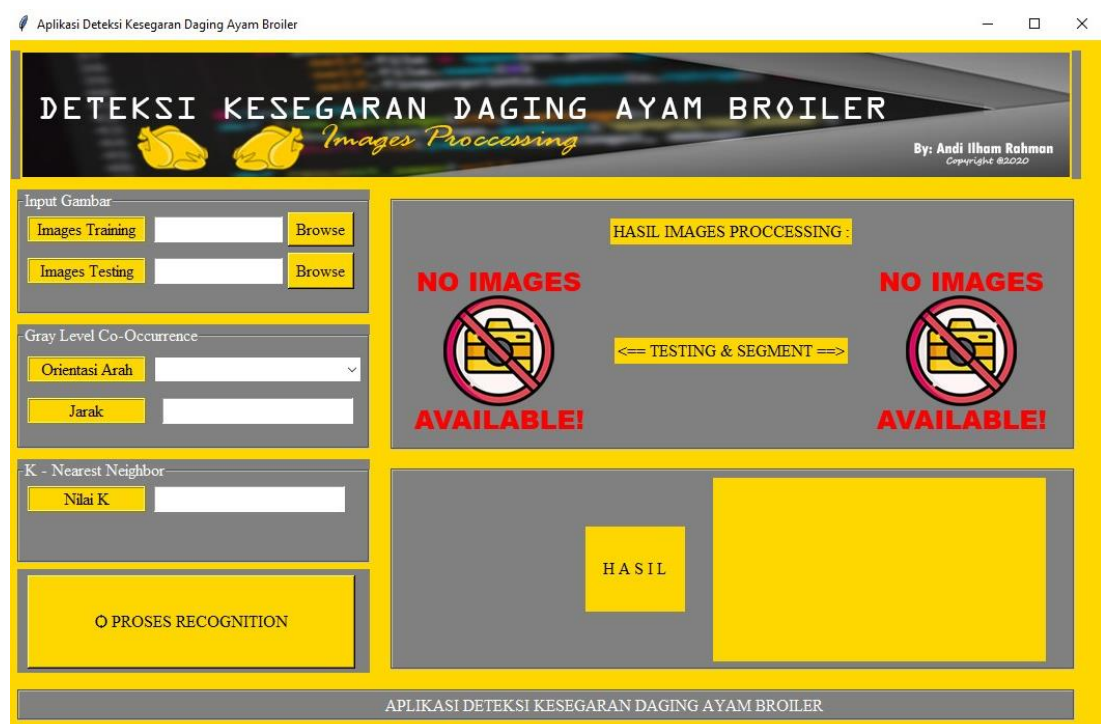
Pada percobaan model K-NN, dengan jumlah $K = 4$, dan Arah GLCM = 45° , dengan jarak = 2, Maka Mendapatkan Nilai Akurasi Sebesar = 90%, dengan Data Training = 90, yaitu data ayam segar = 30, data ayam freezer = 30, dan data ayam busuk = 30, Data Testing = 10.

Pada tahap pengujian metode terdapat kendala pada tahap tingkat akurasi kesegaran daging ayam *broiler* menggunakan tekstur warna, karena penggunaan sampel daging ayam *broiler* harus diperhatikan dari kontras dan posisi dari data sampel, karena ada beberapa sampel data yang tidak bisa dikenali atau tidak terdeteksi. Nilai K sangat berpengaruh untuk akurasi kesegaran daging ayam *broiler* menggunakan tekstur warna, hasil yang di dapatkan untuk metode GLCM dengan K-NN, dengan Nilai K 5 lebih tinggi tingkat akurasi. Hasil yang di dapatkan dari pengujian *Confusion Matrix*. akurasi yang diperoleh sudah hampir sempurna, hasil dari perhitungan akurasi di atas bahwa semakin banyak nilai K semakin tinggi nilai

akurasi yang diperoleh, dan sebaliknya. jumlah K 5 lebih tinggi tingkat akurasi kesegaran daging ayam *broiler* menggunakan tekstur warna, ketimbang jumlah K 1. Dari hasil perhitungan akurasi untuk pengujian Klasifikasi mendapatkan nilai sebesar 90%.

4. 5. Pembahasan Sistem

1. Tampilan Menu Utama



Gambar 4.5 Tampilan Menu Utama

Tampilan pertama saat *program* dijalankan

2. Tampilan *Browse Data Training*

Halaman ini adalah tampilan jika pengguna menekan tombol *Browse* untuk *Data Training*.

3. Tampilan *Browse Data Testing*

Halaman ini adalah tampilan jika pengguna menekan tombol *Browse* untuk *Data Testing*.

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

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

Tampilan ini akan memunculkan nama folder yang telah dipilih pengguna.

5. Tampilan Pemasukan Nilai GLCM

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

6. Tampilan Pemasukan Nilai K-NN

Tampilan ini akan muncul saat pengguna telah memasukan Nilai K.

7. Tampilan Hasil *Images Processing*

Tampilan ini akan muncul saat pengguna mengklik/memilih tombol atau button Proses *Recognition* setelah menginput Data *Traning*, Data *Testing*, Orientasi Arah dan Jarak serta Nilai K. Tampilan ini adalah tampilan hasil pemrosesan.

8. Tampilan Hasil

Tampilan ini akan muncul bersamaan dengan tampilan hasil Deteksi Kesegaran Daging Ayam *Broiler* Menggunakan Tekstur Warna.

4. 6. Hasil Klasifikasi

Tabel 4.5 Perbandingan Nilai

No	Perhitungan Jarak <i>Euclidean Distance</i>	Nilai
1	$d = \sqrt{3.806.4210 + 0.0092 + 0.6456 + 0.0169}$	61.7016
2	$d = \sqrt{11.287.2988 + 0.0107 + 0.6586 + 0.0191}$	106.2449
3	$d = \sqrt{10.262.7435 + 0.0187 + 1.2773 + 0.0474}$	101.3118
4	$d = \sqrt{11.065.4620 + 0.0209 + 1.5018 + 0.0411}$	105.1999
5	$d = \sqrt{30.444.5964 + 0.0017 + 0.6454 + 0.0094}$	174.4856
6	$d = \sqrt{860.4072 + 0.0059 + 1.2029 + 0.0220}$	29.3536
7	$d = \sqrt{134.8362 + 0.0000196 + 0.000000002}$	11.6119
8	$d = \sqrt{1.320.3630 + 0.00000256 + 0.00005184 + 0.00009409}$	36.3368
		
10	$d = \sqrt{2.914.0239 + 0.0024 + 0.1529 + 0.0006}$	53.9831

Tabel 4.5 Hasil Pengurutan Dari Nilai Terkecil Berdasarkan Perhitungan Jarak
Dari Masing-Masing Data *Training* Ke Data *Testing*

No	<i>Euclidean</i>	Y
7	11.6119	Ayam Segar
6	29.3536	Ayam Segar
8	36.3368	Ayam Segar
10	53.9831	Ayam Segar
1	61.7016	Ayam Busuk
3	101.3118	Ayam Busuk
4	105.1999	Ayam Busuk
2	106.2449	Ayam Busuk
5	174.4856	Ayam Busuk

No	<i>Euclidean</i>	Y
7	11.6119	Ayam Segar
6	29.3536	Ayam Segar
8	36.3368	Ayam Segar
10	53.9831	Ayam Segar
1	61.7016	Ayam Busuk
3	101.3118	Ayam Busuk

Jumlah K=6 Yang Telah Ditentukan

Y	Jumlah Data
Ayam Segar	4
Ayam Busuk	2

Tetangga Paling Banyak Muncul = Ayam Segar

4. 7. Evaluasi

Contoh: Jarak 1, orientasi arah sudut 0° dan nilai K 5

Tabel 4.3.1 Hasil Data Testing

Image Testing	Data Aktual	Hasil Klasifikasi
	Ayam Segar	Ayam Freezer
	Ayam Segar	Ayam Segar
	Ayam Segar	Ayam Segar
	Ayam Freezer	Ayam Freezer
	Ayam Freezer	Ayam Freezer
	Ayam Freezer	Ayam Freezer

	Ayam Busuk	Ayam Busuk
	Ayam Busuk	Ayam Busuk
	Ayam Busuk	Ayam Busuk
	Ayam Busuk	Ayam Busuk

Tabel 4.3.1 Hasil Data Uji

	Ayam Segar	Ayam Freezer	Ayam Busuk
Ayam Segar	2	1	0
Ayam Freezer	0	3	0
Ayam Busuk	0	0	4

Jumlah data yang diklasifikasi dengan benar = 9

Jumlah data yang diklasifikasi salah = 1

$$\text{Akurasi} = \frac{\text{Data Uji Dikenali}}{\text{Data uji}} \times 100 \% \quad \text{Akurasi} = \frac{9}{10} \times 100 \% = 90 \%$$

BAB VI

PENUTUP

4. 8. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dan pembahasan yang telah diuraikan sebelumnya maka dapat ditarik kesimpulan bahwa:

Deteksi kesegaran daging ayam *broiler* menggunakan tekstur warna dengan metode *K-Nearest Neighbor* dan fitur ekstraksi *Gray Level Co-Occurrence Matrix* setelah diukur menggunakan *Confusion Matrix* menghasilkan akurasi sebesar 90%.

4. 9. Saran

Setelah melakukan Penelitian deteksi kesegaran daging ayam *broiler* menggunakan tekstur warna, ada beberapa saran yang perlu di perhatikan untuk mencapai tujuan yang di harapkan, yaitu sebagai berikut:

1. Perlu diadakan pengembangan dengan penggunaan metode lain seperti *Gray Level Co-occurrence Matrix* dan *Artificial Neural Network*, *Support Vector Machine* atau metode klasifikasi lainnya sebagai perbandingan untuk mendapatkan hasil terbaik.
2. Untuk pengambilan gambar harus diperhatikan *contras*, *focus*, *background* warna harus polos dan sesuaikan dengan *object*.

DAFTAR PUSTAKA

- [1] Institut Pertanian Bogor.
- [2] Gazali, “Rancang Bangun Alat Pendeteksi Daging Ayam Tiren Menggunakan Sensor Konduktansi Berbasis Mikrokontroler Atmega16,” 2016.
- [3] Computer Vision – Library Binus, Access Sept 13, 2019.
- [4] Amin Faris Mushlihul, “Identifikasi Citra Daging Ayam Berformalin Menggunakan Metode Fitur Tekstur Dan K-Nearest Neighbor (K-NN),” *Jurnal Matematika “Mantik”*, Vol. 04 No. 01, Mei 2018.
- [5] S. Mutrofin, A. Izzah, A. Kurniawardhani And M. Masrur, "Optimasi Teknik Klasifikasi Modified K Nearest Neighbor Menggunakan Algoritma Genetika," Vol. X, No. 1, 2014.
- [6] Husdi, "Pengenalan Ekspresi Wajah Pengguna Elearning Menggunakan Artificial Neural Network Dengan Fitur Ekstraksi Local Binary Pattern Dan Gray Level Co-Occurrence Matrix," *Jurnal Ilmiah Ilkom*, Vol. Viii, No. 3, Pp. 212-213, 2016.
- [7] Asmara Rosa Andrie¹, Puspitasari Dwi², Romlah Siti³, H Qonitatul⁴, Romario Robertus⁵, “Identifikasi Kesegaran Daging Sapi Berdasarkan Citranya Dengan Ekstraksi Fitur Warna Dan Teksturnya Menggunakan Metode Gray Level Cooccurrence Matrix,” Vol. IX, 2017.
- [8] Amin Faris Mushlihul, “Identifikasi Citra Daging Ayam Berformalin Menggunakan Metode Fitur Tekstur Dan K-Nearest Neighbor (K-NN),” *Jurnal Matematika “Mantik”*, Vol. 04 No. 01, Mei 2018.
- [9] Zulfi Faizun Iqbal “Identifikasi Tingkat Kesegaran Daging Sapi Lokal Menggunakan Ekstraksi Fitur Warna Berbasis Gui Matlab,” 2017.
- [10] W Prima Widyawati¹, N Oyas Wahyu², Soesanti Indah³, “Identifikasi Tingkat Kesegaran Daging Ayam Broiler Berdasar Ciri Tekstur Dan Warna Daging,”.
- [11] Sugiarta¹, Agung I Gusti Rai², “Ekstraksi Warna, Tekstur Dan Bentuk Untuk Image Retrieval,” *Seminar Nasional Teknologi Informasi dan Multimedia 2016*, 2016.

- [12] A. Purno and W. Wibowo, "Implementasi Teknik Computer Vision Dengan Metode Colored Markers Trajectory Secara Real Time," vol. VIII, no. 1, 2016.
- [13] Arifin Jaenal¹, Melita Yuliana², "Klasifikasi Jenis Kayu Dengan Gray-Level Co-Occurrence Matrices (Glcms) Dan K-Nearest Neighbor," *Jurnal Ilmiah Teknologi dan Informasi ASIA*, Vol. 7 No. 1, Februari 2013.
- [14] Lihayati Nurul¹, Pawening Ratri Enggar², Furqan Mohammad³, "Klasifikasi Jenis Daging Berdasarkan Tekstur Menggunakan Metode Gray Level Coocurent Matrix," Vol VIII, 2016.
- [15] Dinas Peternakan dan Kesehatan Hewan Kota Gorontalo.
- [16] Arifin Jaenal¹, Melita Yuliana², "Klasifikasi Jenis Kayu Dengan Gray-Level Co-Occurrence Matrices (Glcms) Dan K-Nearest Neighbor," *Jurnal Ilmiah Teknologi dan Informasi ASIA*, Vol. 7 No. 1, Februari 2013.
- [17] Lihayati Nurul¹, Pawening Ratri Enggar², Furqan Mohammad³, "Klasifikasi Jenis Daging Berdasarkan Tekstur Menggunakan Metode Gray Level Coocurent Matrix," Vol VIII, 2016.
- [18] Dinas Peternakan dan Kesehatan Hewan Kota Gorontalo.
- [19] Gazali, "Rancang Bangun Alat Pendeteksi Daging Ayam Tiren Menggunakan Sensor Konduktansi Berbasis Mikrokontroler Atmega16," 2016.
- [20] Syafi'i Slamet Imam¹, Wahyuningrum Rima Tri², dan Muntasa Arif³, "Segmentasi Obyek Pada Citra Digital Menggunakan Metode Otsu Thresholding", Vol. 13, No. 1, Mei 2015.

KODE PROGRAM

1. Pembuatan Form

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

class aplikasiku(tk.Frame):
    def __init__(self, parent):
        tk.Frame.__init__(self)
        self.parent = parent
        self.frames()
        self.form()

    def frames(self):
        #frame judul
        self.FrameAtas = tk.Frame(self.parent, width=1010, height=150,
background="gray", relief='raise')
        self.FrameAtas.grid(row=0, column=0, columnspan=3, padx=4, pady=10,
sticky="NSEW")
        self.FrameAtas.grid_propagate(0)

        #frame input gambar
        self.MainFrame1 = tk.LabelFrame(self.parent, height=120,
background="gray", text="Input Gambar", font="Times", fg="white")
        self.MainFrame1.grid(row=1, column=0, padx=10, sticky="NSEW")
        self.MainFrame1.grid_propagate(0)

        #glcm
        self.MainFrame2 = tk.LabelFrame(self.parent, height=120,
background="gray", text="Gray Level Co-Occurrence", font="Times",
fg="white")
```

```

        self.MainFrame2.grid(row=2,      column=0,      padx=10,      pady=10,
sticky="NSEW")
        self.MainFrame2.grid_propagate(0)

        #knn
        self.MainFrame3      =      tk.LabelFrame(self.parent,      height=100,
background="gray", text="K - Nearest Neighbor", font="Times", fg="white")
        self.MainFrame3.grid(row=3, column=0, padx=10, sticky="NSEW")
        self.MainFrame3.grid_propagate(0)

        #proses
        self.MainFrame4 = tk.Frame(self.parent, height=100, background="gray")
        self.MainFrame4.grid(row=5, column=0, padx=10, pady=6, sticky="NSEW")
        self.MainFrame4.grid_propagate(0)

        #hasil
        self.MainFrame5      =      tk.LabelFrame(self.parent,      background="gray",
font="Times", fg="gray")
        self.MainFrame5.grid(row=1,      column=1,      rowspan=2,      colspan=2,
padx=10, pady=9, sticky="NSEW")
        self.MainFrame5.grid_propagate(0)

        #hasil
        self.MainFrame6      =      tk.LabelFrame(self.parent,      background="gray",
font="Times", fg="gray")
        self.MainFrame6.grid(row=3,      column=1,      rowspan=3,      colspan=2,
padx=10, pady=9, sticky="NSEW")
        self.MainFrame6.grid_propagate(0)

        #kaki
        self.MainFrame7      =      tk.LabelFrame(self.parent,      height=30,      width=1020,
background="gray")
        self.MainFrame7.grid(row=6, column=0, colspan=3, padx=10, pady=10,
sticky="NSEW")
        self.MainFrame7.grid_propagate(0)

    def form(self):

        #frame
        #logo

```

```

self.PathTestingImg ='D:\DATANYA ILHAM\FILE KAMPUS\SEMESTER
8\PROGRAM\WPy64-3810\hasil\sampul.jpg'
self.imT = Image.open(self.PathTestingImg)
self.imT = ImageTk.PhotoImage(self.imT)
self.label1 = tk.Label(self.FrameAtas, width=1010, height=120, bg="gold",
image=self.imT)
self.label1.pack(side="top")

```

```

#frame 1
#label data training
self.label1 = tk.Label(self.MainFrame1, text="Images Training",
font="Times", relief="ridge", width=12, background="gold")
self.label1.grid(row=3, column=0, sticky="NSEW", padx=8, pady=5)

```

```

#textbox data training
self.nilai = tk.StringVar(self.MainFrame1, value="")
self.txtrain = tk.Entry(self.MainFrame1, textvariable = self.nilai, width=20)
self.txtrain.grid(row=3, column=1, sticky="NSEW", pady=5)

```

```

#button browse training
self.path2 = tk.Button(self.MainFrame1, text="Browse", font="Times",
width=6, background="gold",
command=lambda:self.nilai.set(fdialog.askdirectory()))
self.path2.grid(row=3, column=5, padx=5)

```

```

#label data testing
self.label2 = tk.Label(self.MainFrame1, text="Images Testing",
font="Times", relief="ridge", background="gold")
self.label2.grid(row=4, column=0, sticky="NSEW", padx=8, pady=10)

```

```

#textbox data testing
self.nilai1 = tk.StringVar(self.MainFrame1, value="")
self.txtest = tk.Entry(self.MainFrame1, textvariable = self.nilai1)
self.txtest.grid(row=4, column=1, sticky="NSEW", pady=10)

```

```

#button browse testing
self.path1 = tk.Button(self.MainFrame1, text="Browse",
font="Times",background="gold",
command=lambda:self.nilai1.set(fdialog.askopenfilename()))

```

```

self.path1.grid(row=4, column=5, sticky="NSEW", padx=5, pady=5)

#frame 2
#COMBOBOX GLCM
arah = tk.IntVar(self.MainFrame2)
self.label = tk.Label(self.MainFrame2, text="Orientasi Arah", width=12,
font="Times", relief="ridge", background="gold")
self.label.grid(row=5, column=0, padx=8, pady=10)

#arah Orientasi GLCM
arah = tk.StringVar(self.MainFrame2, value="")
self.orientasiarahglcm = ttk.Combobox(self.MainFrame2, textvariable = arah,
width=10)
self.orientasiarahglcm['values']=(0, 45, 90, 135)
self.orientasiarahglcm.current(0)
self.orientasiarahglcm.grid(row=5, column=1, sticky="NSEW", pady=10)

#jarak GLCM
self.label1 = tk.Label(self.MainFrame2, text="Jarak", font="Times",
relief="ridge", width=12, background="gold")
self.label1.grid(row=6, column=0, sticky="NSEW", padx=8, pady=5)

jarak = tk.StringVar(self.MainFrame2, value="")
self.tbjarak = tk.Entry(self.MainFrame2, textvariable = jarak, width=30)
self.tbjarak.grid(row=6, column=1, sticky="NSEW", padx=8, pady=5)

#frame 3
#klasifikasi KNN
#label iterasi
self.label = tk.Label(self.MainFrame3, text="Nilai K", font="Times",
relief="ridge", width=12, background="gold")
self.label.grid(row=0, column=0, sticky="NSEW", padx=8, pady=5)

#textbox iterasi
nilai2 = tk.StringVar(self.MainFrame3, value="")
self.textboxKnn = tk.Entry(self.MainFrame3, textvariable = nilai2, width=30)
self.textboxKnn.grid(row=0, column=1, sticky="NSEW", pady=5)

```

```

#frame 4
#button proses
self.path2 = tk.Button(self.MainFrame4, width=34, height=4, text="O
PROSES RECOGNITION",
                        font="Times",background="gold", command=self.Proses)
self.path2.grid(row=5, column=0, padx=10, pady=6, sticky="NSEW")

```

```

#frame 5
#output
#label hasil image testing
self.label = tk.Label(self.MainFrame5, text="HASIL IMAGES
PROCESSING :", font="Times", background="gold", fg="black")
self.label.grid(row=0, column=0, padx=210, pady=17)

```

```

self.label = tk.Label(self.MainFrame5, text="<== TESTING & SEGMENT
==>", font="Times", background="gold", fg="black")
self.label.grid(row=1, column=0, padx=200, pady=17)

```

```

#images noimages
# self.noimage1 = tk.StringVar(self.MainFrame5, value="D:/DATANYA
ILHAM/FILE KAMPUS/SEMESTER 8/PROGRAM/WPy64-
3810/hasil/noimages.jpg")
self.noimage1 = 'D:/DATANYA ILHAM/FILE KAMPUS/SEMESTER
8/PROGRAM/WPy64-3810/hasil/noimagesavailable!.png'
self.noimageTest1 = Image.open(self.noimage1)
self.noimageTest1 = ImageTk.PhotoImage(self.noimageTest1)
self.labelTest1 = tk.Label(self.MainFrame5, bg = "gray",
image=self.noimageTest1)
self.labelTest1.grid(row=1, column=0, padx=18, pady=1, sticky="W")

```

```

self.noimage2 ='D:/DATANYA ILHAM/FILE KAMPUS/SEMESTER
8/PROGRAM/WPy64-3810/hasil/noimagesavailable!.png'
self.noimageTest2 = Image.open(self.noimage2)
self.noimageTest2 = ImageTk.PhotoImage(self.noimageTest2)
self.labelTest2 = tk.Label(self.MainFrame5, bg = "gray",
image=self.noimageTest2)

```

```

self.labelTest2.grid(row=1, column=0, padx=20, pady=1, sticky="E")

#         self.noimage3 ='D:/DATANYA ILHAM/FILE KAMPUS/SEMESTER
8/PROGRAM/WPy64-3810/hasil/noimages.jpg'
#     self.noimageTest3 = Image.open(self.noimage3)
#     self.noimageTest3 = ImageTk.PhotoImage(self.noimageTest3)
#         self.labelTest3 = tk.Label(self.MainFrame5, bg = "gray",
image=self.noimageTest3)
#     self.labelTest3.grid(row=1, columnspan=80, padx=1, pady=1, sticky="E")

```

```

#frame 6
#label keterangan hasil output
self.value_hasil = tk.StringVar(self.MainFrame6, value="")
self.label12 = tk.Label(self.MainFrame6, textvariable=self.value_hasil,
width=35, height=9, font="Times", background="gold")
self.label12.grid(row=1, column=0, columnspan=20, padx=309, pady=7)

```

```

self.label12 = tk.Label(self.MainFrame6, text="LABEL AKURASI",
width=30,height=4, font="Times", background="gold", fg="black")
self.label12.grid(row=1, column=0, columnspan=5, padx=25, pady=50)

```

```

#frame 7
#label kaki
self.label = tk.Label(self.MainFrame7, text="APLIKASI DETEKSI
KESEGARAN DAGING AYAM BROILER", font="Times", fg="white")
self.label.grid(row=2, column=2, sticky="NSEW", padx=350, pady=1)
self.label.config(background="gray")
self.MainFrame7.grid_propagate(0)

```

```

def Proses(self):

```

```

    #get images training
    folder_train = self.txtrain.get()

```

```

    #get images testing
    folderTesting = self.txtest.get()

```

```

#get arah Orientasi GLCM
arahGLCM = self.orientasiarahglcm.get()

#jarak GLCM
jarakGLCM = self.tbjarak.get()

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

#hasil images processing
#images 1
a = analisis_citra.analisis_citra()
hasil, img_segmen = a.proses_rekognisi(folder_train, folderTesting,
arahGLCM, jarakGLCM, NilaiK)
self.value_hasil.set(hasil)
img = Image.open(self.txttest.get())
img = img.resize((165, 165), Image.ANTIALIAS)
imgori = ImageTk.PhotoImage(img)
self.labelTest1.configure(image=imgori)
self.labelTest1.image_names = imgori

#images 2
img2 = cv2.cvtColor(img_segmen, cv2.COLOR_BGR2RGB)
img2 = Image.fromarray(img2)
img2 = img2.resize((165,165), Image.ANTIALIAS)
img_segment = ImageTk.PhotoImage(img2)
self.labelTest2.configure(image=img_segment)
self.labelTest2.image_names = img_segment

def main():
    root = tk.Tk()
    root.geometry('1060x670+130+10')
    root.configure(background='gold')
    app = aplikasiku(root)
    app.parent.title("Aplikasi Deteksi Kesegaran Daging Ayam Broiler")
    root.mainloop()

if __name__ == '__main__':
    main()

```

2. Image Processing

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

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

    def fitur_glcmm(self, img, arah, Jarak):
        img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        matrix_glcmm = feature.greycomatrix(img, [Jarak], [arah], symmetric=False,
normed=True)
        contrast = feature.greycomprops(matrix_glcmm, 'contrast')[0,0]
        homogeneity = feature.greycomprops(matrix_glcmm, 'homogeneity')[0,0]
        ent = entropy(matrix_glcmm.flatten())
        energy = feature.greycomprops(matrix_glcmm, 'energy')[0,0]
        hasil_fitur = [contrast, homogeneity, ent, energy]
        return hasil_fitur

    def get_Img_Path(self, folderPath):
        imPath = []
```

```

    for path in paths.list_images(folderPath):
        imPath.append(path)
    return imPath

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

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

def testing_data(self, img, arah, Jarak):
    data_test = []
    img = cv2.imread(img)
    hasil_segmen = self.segmentasi(img)
    hasil_ekstrak_glcmm = self.fitur_glcmm(hasil_segmen, arah, Jarak)
    data_test.append(hasil_ekstrak_glcmm)
    return data_test, hasil_segmen

def proses_rekognisi(self, folder_train, image_test, arah, jarak, NilaiK):
    x_train, y_train = self.training_data(folder_train, arah, jarak)
    x_test, hasil_segmen = self.testing_data(image_test, arah, jarak)
    clf = KNeighborsClassifier(n_neighbors=int(NilaiK), weights='uniform',
algorithm='auto', metric='euclidean')
    clf.fit(x_train, y_train)
    final_hasil = clf.predict(x_test)
    return final_hasil, hasil_segmen

```

No Gambar	Data Gambar	Con(X1)	Hom(X2)	Ent(X3)	Ene(X4)	Y
0		183.083875	0.62069490	4.85224623	0.37142602	Ayam
		34635682	27834016	7501204	186123896	Busuk
1		227.629361	0.61352745	4.86033754	0.36325627	Ayam
		78151085	61092432	56548585	581857667	Busuk
2		222.692802	0.58020078	5.17895109	0.28378781	Ayam
		98257308	35920091	01874305	545566073	Busuk
3		226.580115	0.57248088	5.27426237	0.29874907	Ayam
		13613428	82219949	9886439	481215823	Busuk
4		295.871415	0.75867761	3.24534072	0.59863436	Ayam
		83941417	00805026	02590134	29282341	Busuk
5		159.824208	0.49860361	5.93615955	0.19322147	Ayam
		13623932	905340944	5201732	71131554	Busuk

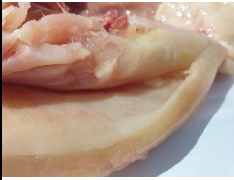
6		254.215805 3513366	0.57164312 82970126	5.21417359 4948957	0.30730824 913185256	Ayam Busuk
7		247.403512 24767323	0.70021259 58552593	3.97403047 7631664	0.48174434 78860245	Ayam Busuk
8		204.747432 43172313	0.60422403 8572193	5.19326427 9347938	0.28678807 965499886	Ayam Busuk
9		272.523138 22177086	0.58664074 98535273	5.30931857 9561377	0.26335743 99409593	Ayam Busuk
10		202.600644 5254306	0.64517192 89649958	4.74101997 689625	0.39356384 73851597	Ayam Busuk
11		160.948523 9394692	0.63287408 11051097	4.54333935 4904578	0.40687459 182160957	Ayam Busuk

12		272.519218 77932796	0.63046692 50146908	4.61240957 8617767	0.39407978 625204465	Ayam Busuk
13		291.400272 06477856	0.57802643 440331	5.10700036 6167604	0.31879337 503200944	Ayam Busuk
14		181.795912 20291422	0.60642273 61979582	4.88876135 0386519	0.34069280 627021903	Ayam Busuk
15		256.865521 9502559	0.64916099 82447608	4.44887751 5517952	0.40452919 53536868	Ayam Busuk
16		210.217377 32966346	0.55895990 00586074	5.51686165 9368665	0.25128485 417487556	Ayam Busuk
17		122.026019 7424983	0.54564812 62725484	5.39034842 0891471	0.25482483 134445705	Ayam Busuk

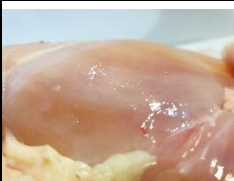
18		159.930874 12966247	0.53969992 91742122	5.63899893 8426629	0.26111570 314596844	Ayam Busuk
19		205.686606 62602937	0.57618140 6640074	5.27816747 7444051	0.29275471 626097466	Ayam Busuk
20		168.492519 12099226	0.58190169 2405178	5.22476266 7454868	0.33162365 058175663	Ayam Busuk
21		111.608386 2210252	0.64927621 69822972	4.53812269 95264585	0.43654356 81722376	Ayam Busuk
22		137.892621 534956	0.55733247 83323148	5.51683494 9932977	0.26135300 34945276	Ayam Busuk
23		139.182527 3771131	0.56741287 10757882	5.42479758 6894569	0.27524872 39189219	Ayam Busuk

24		134.364895 1118252	0.56779507 88903159	5.45692909 0930548	0.31658005 5752846	Ayam Busuk
25		111.341241 08285774	0.61358733 3229637	5.03906003 4943443	0.38535805 57856221	Ayam Busuk
26		151.424081 45704235	0.63939041 30301815	4.69089468 3573584	0.42251879 801079517	Ayam Busuk
27		179.260225 72746416	0.69539060 42440727	3.94138790 17075786	0.49914052 45531439	Ayam Busuk
28		170.086946 45310607	0.57241997 16937055	5.21892527 5420364	0.31079420 91347573	Ayam Busuk
29		103.256838 20759488	0.61341860 3091376	4.81422528 2720978	0.37534853 75682902	Ayam Busuk

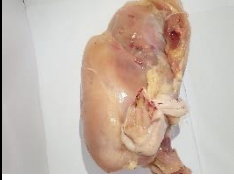
30		57.5502181 5705181	0.64250177 44363066	4.73389301 5256731	0.34212507 286717403	Ayam Freezer
31		171.235114 98338357	0.74771029 96159988	3.66603801 94922157	0.54109411 36056646	Ayam Freezer
32		251.029150 97641483	0.61127026 35769072	4.93043303 9154945	0.26744707 21854021	Ayam Freezer
33		159.376069 9573265	3.70536491 97290772	0.74198093 05914356	0.52417422 77003811	Ayam Freezer
34		73.0775153 0418639	0.75331365 38501363	4.14473394 5834285	0.43283131 098611205	Ayam Freezer
35		53.3762957 9254097	0.52913821 50916554	6.01336947 8794227	0.20416168 61367493	Ayam Freezer

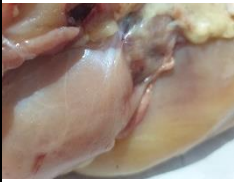
36		52.0852367 0559523	0.86273954 2439376	2.02304434 79661884	0.78389011 96800815	Ayam Freezer
37		84.7452776 0821819	0.70655293 53379878	4.05521036 2005277	0.50917722 85235403	Ayam Freezer
38		101.535955 57429361	0.49370441 621166367	6.31025890 8234691	0.17522352 032959077	Ayam Freezer
39		85.7909726 4060677	0.76425923 15691265	3.40419028 9267165	0.59931820 73531197	Ayam Freezer
40		142.821022 2775033	0.56715862 92733113	5.66845147 7093422	0.26758475 005912624	Ayam Freezer
41		96.7106067 8043648	0.61819116 50850724	4.93550570 6487535	0.38550395 35436671	Ayam Freezer

42		65.8942687 1486276	0.57298451 55584715	5.26369826 5741031	0.32926057 502864414	Ayam Freezer
43		133.575724 61429797	0.58073859 9064967	5.20021429 286845	0.31543483 98592906	Ayam Freezer
44		62.3343948 81467595	0.59630252 78066716	5.03352165 019261	0.30080196 38842133	Ayam Freezer
45		63.7043652 4343745	0.69255637 17443884	4.00148472 2667032	0.48445873 42847625	Ayam Freezer
46		64.8693253 115078	0.59351517 89848028	5.25605581 148869	0.29437599 916751833	Ayam Freezer
47		117.168144 6304369	0.67895220 4956381	4.23408389 3883331	0.44121945 6899103	Ayam Freezer

48		100.482233 23383901	0.70922256 9982951	4.02740664 0264625	0.49256606 945052245	Ayam Freezer
49		101.741657 16687739	0.56271445 02458076	5.44049281 86194415	0.25877094 49960625	Ayam Freezer
50		103.913115 32055151	0.65478961 16702188	4.52093075 992556	0.42844285 03074426	Ayam Freezer
51		129.146072 55082637	0.63475646 03637892	4.77551484 8544503	0.35693491 061522176	Ayam Freezer
52		169.623547 47463036	0.61924652 3459508	4.77623630 1162345	0.35861173 84749408	Ayam Freezer
53		129.632400 15540933	0.57077454 9854626	5.26431275 1348588	0.28531716 611928026	Ayam Freezer

54		67.3222509 8410598	0.57505704 69493427	5.50424387 4468532	0.23537114 95905097	Ayam Freezer
55		122.136900 2415039	0.52985755 90778603	5.80360151 0730162	0.19979450 09796675	Ayam Freezer
56		149.125927 17287592	0.61956003 40108194	4.95467548 9155983	0.35128342 72902398	Ayam Freezer
57		73.0187792 3684273	0.66808881 27460555	4.28400641 4542658	0.44694977 50627066	Ayam Freezer
58		63.5560336 49825935	0.59085545 48473735	5.00712048 70349435	0.30524052 0508641	Ayam Freezer
59		174.416396 86497444	0.49151053 99800873	6.14470471 4796075	0.18305346 38721079	Ayam Freezer

60		101.307793 96702388	0.87474996 81431656	1.88494822 9927025	0.79486630 55038493	Ayam Segar
61		70.1054616 8976151	0.66460714 0896686	4.36191812 4241425	0.47580255 666491245	Ayam Segar
62		56.3096407 1107648	0.77106186 22028029	3.10293571 36161586	0.64326751 15138662	Ayam Segar
63		248.358241 3051036	0.57419663 05792455	5.18995265 2680271	0.29540949 75920056	Ayam Segar
64		165.155369 1221118	0.76284284 96509057	3.24267616 25447167	0.61951834 06274495	Ayam Segar
65		94.5191282 5810123	0.66279012 67111552	4.59608276 9329412	0.41350997 72173168	Ayam Segar

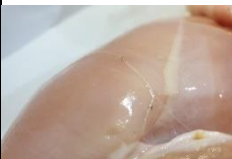
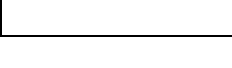
66		56.5542025 1828721	0.56880945 39156874	5.43409244 7682234	0.29494118 42520133	Ayam Segar
67		68.7499708 771571	0.68385075 61881454	4.31992647 61352465	0.47385234 875376653	Ayam Segar
68		94.5884578 8722061	0.74545024 12972164	3.53031482 0037996	0.57154535 21310413	Ayam Segar
69		179.567799 70112592	0.75370564 17794841	3.38684461 6968867	0.57982722 04131929	Ayam Segar
70		119.602512 32511533	0.58498279 46849522	5.33244618 677517	0.29209654 87982777	Ayam Segar
71		120.758250 62445942	0.70279913 00703821	4.03338328 7441564	0.48492760 777643223	Ayam Segar

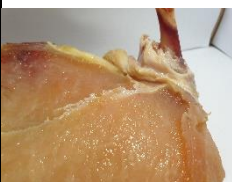
72		138.109226 63542294	0.62081311 41535986	4.85952323 03049585	0.35582014 69984233	Ayam Segar
73		212.169212 69210908	0.72462469 88312849	3.64561723 91724375	0.52929200 63673272	Ayam Segar
74		180.912851 9844224	0.58249154 2057091	5.32004702 836505	0.26821320 037752855	Ayam Segar
75		152.030559 62455	0.60209605 70698016	5.04844817 2123798	0.31582639 36772833	Ayam Segar
76		300.241534 68686484	0.51041550 58104882	5.81906739 3566394	0.25377881 57575237	Ayam Segar
77		188.029580 78077694	0.55247356 02747726	5.40002029 805844	0.29217218 19480545	Ayam Segar

78		85.2367786 434192	0.65705818 07277591	4.50369027 911542	0.41625408 894075056	Ayam Segar
79		185.418502 21406165	0.68605652 01282213	4.27940128 9602164	0.46262129 73146941	Ayam Segar
80		126.740951 16353558	0.65725428 20693581	4.45450605 5697201	0.41641254 345578943	Ayam Segar
81		163.458891 01526672	0.55609678 6607064	5.47102973 9682734	0.28859652 14319003	Ayam Segar
82		107.999681 78166831	0.67088208 21092325	4.39049555 8282348	0.41805780 653409347	Ayam Segar
83		152.884697 00430142	0.53613594 35498185	5.76336565 936104	0.18821808 674190835	Ayam Segar

84		74.1029913 5621729	0.71245999 90532665	3.85611115 44780133	0.54674444 12152471	Ayam Segar
85		150.720306 86018707	0.79398901 27133541	2.95191892 69436423	0.65020459 11690653	Ayam Segar
86		109.775777 71614469	0.71846179 8310835	4.08580216 2433137	0.48534663 778806897	Ayam Segar
87		85.0508068 9820198	0.71545041 1042428	4.04156539 6216265	0.49196716 517678896	Ayam Segar
88		104.013994 14421601	0.83273361 62466248	2.56536373 013787	0.70381174 9449142	Ayam Segar
89		175.369373 33740468	0.76652098 67492531	3.65767795 1575042	0.52614112 66166133	Ayam Segar

Hasil Perhitungan Data *Testing*

No Gambar	Data Gambar	Con(X1)	Hom(X2)	Ent(X3)	Ene(X4)	Y
0		121.387651	0.71705994	4.04873401	0.50163868	?
		39160904	65209052	813019	66559918	
						
1		87.2279774	0.72626790	3.88861182	0.51803312	?
		674355	60762323	6374533	68013488	
						
2		85.3320467	0.80959805	3.27494103	0.58908389	?
		4577536	5797875	738266	44650335	
						
3		127.103468	0.69316040	4.07201680	0.45663793	?
		78490643	40085572	44370695	781299256	
						
4		57.4527410	0.63092910	4.85911178	0.31523348	?
		1594856	27906105	9291147	52944615	
						
5		58.4663189	0.70573824	4.06664407	0.47687653	?
		7932467	12960994	9341159	621954335	
						

6		242.660925 52887096	0.58383183 01660271	5.24212668 4700891	0.28143186 25337788	?
7		208.692541 36920375	0.58191327 0079425	5.11040513 0802241	0.33203086 22930639	?
8		159.975656 85099073	0.53951577 0949142	5.53388842 6850323	0.25558424 686838366	?
9		300.065902 2863812	0.64536657 97068796	4.62666141 9128022	0.38568882 60926947	?

DAFTAR RIWAYAT HIDUP



Nama : Andi Moh. Ilham Zulkarnain Rahman
Tempat, Tgl Lahir : Gorontalo, 23 Maret 1998
Pekerjaan : Mahasiswa
Email : andiilham551@gmail.com

Riwayat Pendidikan :

1. Bulan Juni Tahun 2004, telah menyelesaikan Pendidikan di Taman Kanak Kanak Raudatul Jannah 2, Kecamatan Kota Utara, Kota Gorontalo
2. Bulan Juni Tahun 2010, telah menyelesaikan Pendidikan di Sekolah Dasar Negeri 86, Kecamatan Kota Tengah, Kota Gorontalo
3. Bulan Juni Tahun 2013, telah menyelesaikan Pendidikan di Sekolah Menengah Pertama Negeri 6 Gorontalo, Kecamatan Kota Selatan, Kota Gorontalo
4. Bulan Mei Tahun 2016, telah menyelesaikan Pendidikan di Sekolah Menengah Kejuruan Negeri 1 Gorontalo, Kecamatan Sipatana, Kota Gorontalo
5. Bulan Mei Tahun 2016, telah di terima menjadi Mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo, Kecamatan Dulalowo Timur, Kota Gorontalo

Pengalaman Kerja :

1. Bulan Agustus Tahun 2016 – Bulan Februari Tahun 2017, bekerja di Delima Studio Entertainment sebagai *Editor Photo* dan *Photography*
2. Bulan Februari Tahun 2017 – Bulan Agustus Tahun 2019, berkerja di Satu Delapan *Computer* sebagai *IT-Teknisi Hardware* dan *Software*
3. Bulan Juni Tahun 2019 – Sekarang, Wirausaha di Rahayu Mandiri Tech sekaligus *Freelance Job Project Multimedia, Service Hardware* dan *Software PC Computer/Laptop*



**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS ICHSAN
(UNISAN) GORONTALO**

SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT REKOMENDASI BEBAS PLAGIASI

No. 0317/UNISAN-G/S-BP/IV/2020

Yang bertanda tangan di bawah ini :

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

Dengan ini Menyatakan bahwa :

Nama Mahasisw : ANDI MOH. ILHAM ZULKARNAIN RAHMAN
NIM : T3116042
Program Studi : Teknik Informatika (S1)
Fakultas : Fakultas Ilmu Komputer
Judul Skripsi : Deteksi Kesegaran Daging Ayam Broiler Menggunakan
Tekstur Warna Dengan Metode Gray Level
Co-Occurrence Matrix Dan K-Nearest Neighbor

Sesuai dengan hasil pengecekan tingkat kemiripan skripsi melalui aplikasi Turnitin untuk judul skripsi di atas diperoleh hasil Similarity sebesar 32%, berdasarkan SK Rektor No. 237/UNISAN-G/SK/IX/2019 tentang Panduan Pencegahan dan Penanggulangan Plagiarisme, bahwa batas kemiripan skripsi maksimal 35% dan sesuai dengan Surat Pernyataan dari kedua Pembimbing yang bersangkutan menyatakan bahwa isi softcopy skripsi yang diolah di Turnitin SAMA ISINYA dengan Skripsi Aslinya serta format penulisannya sudah sesuai dengan Buku Panduan Penulisan Skripsi, untuk itu skripsi tersebut di atas dinyatakan BEBAS PLAGIASI dan layak untuk diujikan.

Demikian surat rekomendasi ini dibuat untuk digunakan sebagaimana mestinya.

Gorontalo, 07 Juli 2020
Tim Verifikasi,



Sunarto Taliki, M.Kom
NIDN. 0906058301

Tembusan :

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

DETEKSI KESEGARAN DAGING AYAM BROILER MENGUNAKAN TEKSTUR WARNA DENGAN METODE GRAY LEVEL CO-OCCURRENCE MATRIX DAN K-NEAREST NEIGHBOR

ORIGINALITY REPORT

32%

SIMILARITY INDEX

29%

INTERNET SOURCES

6%

PUBLICATIONS

18%

STUDENT PAPERS

PRIMARY SOURCES

1

www.scribd.com

Internet Source

8%

2

media.neliti.com

Internet Source

5%

3

pt.slideshare.net

Internet Source

2%

4

ejournal.iaingawi.ac.id

Internet Source

2%

5

Submitted to Universitas Dian Nuswantoro

Student Paper

2%

6

jurnal.unsil.ac.id

Internet Source

2%

7

Submitted to Higher Education Commission
Pakistan

Student Paper

1%

ojs.amikom.ac.id

8	Internet Source	1%
9	e-journals.unmul.ac.id Internet Source	1%
10	pt.scribd.com Internet Source	1%
11	stmikplk.ac.id Internet Source	1%
12	ejournal.raharja.ac.id Internet Source	1%
13	etheses.uin-malang.ac.id Internet Source	1%
14	es.scribd.com Internet Source	1%
15	ojs.uniska-bjm.ac.id Internet Source	1%
16	id.dinus.ac.id Internet Source	1%
17	join.if.uinsgd.ac.id Internet Source	1%
18	jurnal.stikom.edu Internet Source	<1%
19	jurnalsaintek.uinsby.ac.id Internet Source	<1%

20	ilhamnurhadi95.blogspot.com Internet Source	<1 %
21	Submitted to Universitas Brawijaya Student Paper	<1 %
22	Budy Santoso, Azminuddin I. S. Azis, Andi Bode. "Pengendalian Lampu Lalu Lintas Cerdas di Persimpangan Empat Ruas yang Kompleks Menggunakan Algoritma Adaptive Neuro Fuzzy Inference System", Jurnal Edukasi dan Penelitian Informatika (JEPIN), 2020 Publication	<1 %
23	sir.stikom.edu Internet Source	<1 %
24	ejournal.nusamandiri.ac.id Internet Source	<1 %

Exclude quotes On
Exclude bibliography On

Exclude matches < 25 words