

Implementasi Pengolahan Citra dengan Menggunakan Metode K-Nearest Neighbor Untuk Mengetahui Daging Ayam Busuk dan Daging Ayam Segar

Meanus Laia, Rivalry K Hondro, Taronisokhi Zebua

Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: meanus.laia@gmail.com

Submitted 06-02-2021; Accepted 10-04-2021; Published 25-04-2021

Abstrak

Daging hewan adalah otot hewan yang tersusun dari serat-serat yang sangat kecil, masing-masing berupa sel memanjang yang disatukan oleh jaringan ikat, membentuk berkas ikatan yang pada kebanyakan daging jelas kelihatan lemak pembuluh darah dan urat syaraf. Meningkatnya konsumsi daging hewan, maka tidak heran daging-daging konsumsi banyak ditemukan di pasar. Pada dasarnya daging-daging konsumsi tersebut dijual oleh pedagang sesuai dengan jenisnya. Jenis daging konsumsi yang sering dicampur adalah daging ayam busuk dan daging ayam segar. Konsumen banyak yang tidak menyadari tindakan pencampuran daging konsumsi tersebut karena secara kasat mata daging hewan sulit untuk dibedakan oleh konsumen. Pencampuran daging tersebut sangat merugikan konsumen. Penelitian ini membahas tentang klasifikasi daging ayam busuk dan daging ayam segar menggunakan metode K-Nearest Neighbor. Bahan citra yang digunakan pada penelitian ini adalah daging ayam busuk dan daging ayam segar. Sampel gambar asli disimpan dalam RGB (Red Green Blue). Tahap pertama yaitu merubah ukuran gambar menjadi 5x5 pixel dan dikonversi menjadi keabu-abuan (Grayscale). Untuk mengenali ciri suatu gambar, digunakan metode Gray Level Co-Occurrence Matrix (GLCM). Metode K-Nearest Neighbor pada penelitian ini menggunakan nilai $k = 3$. Orientasi sudut yang digunakan 0° . Akurasi tertinggi didapatkan pada pengujian pada citra daging ayam segar.

Kata Kunci: Daging; GLCM; Klasifikasi; K-Nearest Neighbor; Akurasi

Abstract

Animal flesh is animal muscle composed of very small fibers, each consisting of elongated cells held together by connective tissue, making bonds that generally contain clear flesh, fat, blood, and nerve fibers. Increased consumption of animal meat, it is no wonder meat consumption is found in many markets. Basically meat - meat consumption is sold by traders according to its type. Types of meat that are often consumed are rotten chicken meat and fresh chicken meat. Many consumers do not like the act of mixing meat consumption because it is difficult to distinguish by consumers. This meat mixer is very detrimental to consumers. This study discusses rotten chicken meat and rotten chicken meat using the K-Nearest Neighbor method. The image material used in this study was rotten chicken meat and fresh chicken meat. The original image sample is saved in RGB (Red Green Blue). The first step is resizing the image to 5x5 pixels and turning grayish (Grayscale). To question the characteristics of the image, the Gray Level Co-Occurrence Matrix (GLCM) method is used. The K-Nearest Neighbor method in this study uses the value $k = 3$. The angle orientation used is 0° . The highest accuracy obtained in testing the image of fresh chicken meat.

Keywords: Meat; GLCM; K-Nearest Neighbor; Classification; Accuracy

1. PENDAHULUAN

Pengolahan citra adalah setiap bentuk pengolahan sinyal dimana input adalah gambar, seperti foto atau video bingkai, sedangkan *output* dari pengolahan gambar dapat berupa gambar atau sejumlah karakteristik atau parameter yang berkaitan dengan gambar[1]. Kebanyakan gambar teknik pemrosesan melibatkan atau memperlakukan foto sebagai dimensi dua sinyal dan menerapkan standar teknik pemrosesan sinyal untuk itu, biasanya hal tersebut mengacu pada pengolahan gambar digital tetapi dapat juga digunakan untuk optik dan pengolahan gambar *analog*. Akuisisi gambar atau yang menghasilkan gambar *input* di tempat pertama disebut sebagai pencitraan.

Daging hewan adalah otot hewan yang tersusun dari serat-serat yang sangat kecil, masing-masing berupa sel memanjang yang disatukan oleh jaringan ikat, membentuk berkas ikatan yang pada kebanyakan daging jelas kelihatan lemak pembuluh darah dan urat syaraf [2]. Bila potongan daging diamati secara teliti maka tampak dengan jelas bahwa daging terdiri atas tenunan yang terdiri atas air, protein, tenunan lemak dan potongan tulang.

Meningkatnya konsumsi daging ayam, maka tidak heran daging-daging konsumsi banyak ditemukan di pasar. Pada dasarnya daging-daging konsumsi tersebut dijual oleh pedagang sesuai dengan jenisnya, namun tidak sedikit pedagang melakukan pencampuran daging konsumsi untuk mendapatkan keuntungan yang besar. Jenis daging konsumsi yang sering dicampur adalah daging ayam busuk dan daging ayam segar. Konsumen banyak yang tidak menyadari tindakan pencampuran daging ayam konsumsi tersebut karena daging ayam terlalu banyak, sehingga sulit untuk dibedakan oleh konsumen. Pencampuran daging tersebut sangat merugikan konsumen.

Seperti kita ketahui kemajuan teknologi dibidang pengolahan citra digital sudah berkembang, tepatnya teknik pengenalan pola suatu citra digital sehingga digunakan pengolahan citra digital untuk mengklasifikasikan kelas daging. Penelitian ini adalah tahapan awal untuk merancang perangkat lunak yang dapat mengklasifikasikan jenis daging ayam busuk dan daging ayam segar, menguji metode tersebut untuk dapat menghasilkan teknik klasifikasi dengan akurasi yang baik.

Salah satu cara untuk mengetahui daging ayam busuk dan daging ayam segar di bidang informatika adalah menggunakan pengolahan citra. Berdasarkan penelitian ini akan dibuat suatu sistem pengolahan citra untuk mencari nilai daging ayam busuk dan daging ayam segar menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM), dan mengklasifikasikan dengan menggunakan metode K-Nearest Neighbor (KNN) dengan tujuan untuk mengetahui nilai akurasi dari hasil citra yang dideteksi.

Pemilihan metode *Gray Level Co-occurrence Matrix* (GLCM) adalah untuk mengetahui ciri tekstur metode yang hanya menggunakan 4 fitur mampu mencapai akurasi klasifikasi tertinggi dibandingkan metode filter gabor[3]. Metode Klasifikasi *K-Nearest Neighbor* (K-NN) digunakan untuk mengklasifikasi data karena Klasifikasi *K-Nearest Neighbor* (K-NN) mencapai hasil akurasi lebih tinggi dari pada klasifikasi lainnya. Sampel daging yang akan dikenali adalah daging ayam busuk dan daging ayam segar untuk bagian daging paha dan dada. Data diambil ke lapangan yaitu pasar daging ayam, menggunakan kamera *handphone*. Data dimasukkan ke sistem melalui tahap training dan tahap testing.

2. METODOLOGI PENELITIAN

2.1 Pengolahan Citra Digital

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer. Secara matematis, citra merupakan fungsi kontinu (*continue*) dengan intensitas cahaya pada bidang dua dimensi. Reperesentasi dari fungsi kontinu menjadi nilai-nilai diskrit disebut digitalisasi citra. Sebuah citra digital dapat diwakili oleh sebuah matriks dua dimensi $f(x,y)$ yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut piksel (*pixel = picture element*) atau elemen terkecil dari sebuah citra [2].

2.2 Kalsifikasi

Klasifikasi adalah proses penemuan model (atau fungsi) yang menggambarkan dan membedakan kelas data atau konsep yang bertujuan agar bisa digunakan untuk memprediksi kelas dari objek yang label kelasnya tidak diketahui. Klasifikasi data terdiri dari 2 langkah proses. Pertama adalah *learning* (fase *training*), dimana algoritma klasifikasi dibuat untuk menganalisa data *training* lalu direpresentasikan dalam bentuk *rule* klasifikasi. Proses klasifikasi didasarkan pada empat komponen [3]:

1. *Kelas*
 Variabel dependen yang berupa kategorikal yang merepresentasikan 'label' yang terdapat pada objek. Contohnya: resiko penyakit jantung, resiko kredit, *customer loyalty*, jenisgempa.
2. *Predictor*
 Variabel independen yang direpresentasikan oleh karakteristik (atribut) data. Contohnya: merokok, minum alkohol, tekanan darah, tabungan, aset, gaji.
3. *Training dataset*
 Satu set data yang berisi nilai dari kedua komponen di atas yang digunakan untuk menentukan kelas yang cocok berdasarkan *predictor*.
4. *Testing dataset*
 Berisi data baru yang akan diklasifikasikan oleh model yang telah dibuat dan akurasi klasifikasi dievaluasi [3].

2.3 Daging Ayam

Ayam merupakan salah satu ternak unggas yang sudah tidak asing lagi dikalangan masyarakat. Ayam yang digunakan oleh masyarakat untuk diolah biasanya adalah ayam potong, untuk memilih daging ayam segar yang biasa perlu diperhatikan beberapa hal, yaitu warna daging yang putih kekuningan, warna lemak yang putih kekuningan dan merata di bawah kulit, memiliki bau yang segar, kekenyalan yang elastis dan tidak ada tanda-tanda memar atau tanda lain yang mencurigaka. Daging ayam termasuk mengandung gizi yang tinggi, selain dari proteinnya juga daging ayam mengandung lemak. Protein pada ayam yaitu 18,2g sedangkan lemaknya berkisar 25,0 g [4].

2.4 K-Nearest Neighbor

K-Nearest Neighbor adalah metode yang bersifat *supervised*, di mana hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas kategori pada K-NN. K-NN adalah sebuah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumnya. Termasuk dalam *supervised learning*, di mana hasil *query instance* yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori yang ada dalam K-NN [8].

Rumus K-NN :

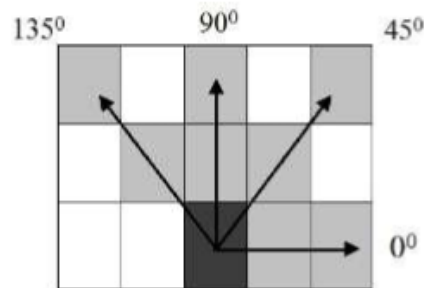
$$d(x_i, x_j) = \sqrt{\sum_{k=1}^n (a_r(x_i) - a_r(x_j))^2} \quad (1)$$

Keterangan :

- $d(x_i, x_j)$: Jarak *Euclidean* (*Euclidean Distance*)
 $(x_i), (x_j)$: *record* ke-i, *record* ke-j
 (a_r) : data ke-r
 i,j : 1,2,3,..n
 n : dimensi objek [10].

2.5 Metode Gray Level Co-occurrence Matrix (GLCM)

Gray Level Co-occurrence Matrix (GLCM) pertama kali diusulkan oleh Haralick pada tahun 1979 dengan 28 fitur untuk menjelaskan pola spasial [5]. Gray Level Co-occurrence Matrix (GLCM) adalah suatu matriks yang elemen-elemennya merupakan jumlah pasangan piksel yang memiliki tingkat kecerahan tertentu, di mana pasangan piksel itu terpisah dengan jarak d , dan dengan suatu sudut inklinasi θ . Dengan kata lain, matriks kookurensi adalah probabilitas munculnya *gray level* i dan j dari dua piksel terpisah pada jarak yaitu d dan sudut dilambangkan θ . Suatu piksel yang bertetangga memiliki d diantara keduanya, dapat terletak didelapan arah berlainan.



Gambar 1. Hubungan ketetangaan antar piksel dan jarak spasial

Arah piksel tetangga untuk mewakili jarak dapat dipilih, misalnya 135° , 90° , 45° , 0° . Berikut tahapan ekstraksi fitur GLCM :

1. Normalisasi, yaitu proses mengubah rentang nilai intensitas *pixel*.
2. Membuat tabel area kerja matriks.
3. Menentukan piksel ketetangga spasial antara piksel referensi dengan piksel tetangganya, yaitu menentukan nilai sudut θ dan jarak d .
4. Menghitung jumlah nilai kookurensi dan mengisikannya pada area kerja yang telah dibuat.
5. Menormalisasi masing-masing nilai matriks untuk mengubahnya ke bentuk probabilitas.
6. Melakukan ekstraksi fitur.

Beberapa jenis ciri tekstur yang dapat diekstraksi dengan matriks kookurensi [11]. Beberapa diantaranya adalah sebagai berikut :

1. Contrast

Hasil perhitungan nilai yang berkaitan dengan jumlah keberagaman intensitas keabuan citra.

$$\text{Contrast} = \sum_{i,j} (i - j)^2 P_d(i, j) \quad (2)$$

2. Energy

Energy adalah pengukuran tekstur dari gambar *grayscale* yang merepresentasikan perubahan, merefleksikan keseragaman berat dan tekstur distribusi gambar *grayscale*, yang menggunakan persamaan 9.

$$\text{Energy} = \sum_{i,j} P(i, j)^2 \quad (3)$$

3. Entropy

Menunjukkan ketidakteraturan ukuran bentuk. Entropi mengukur informasi atau pesanyang hilang dari sabuah sinyal tranmisi juga menghitung informasi gambar.

$$\text{Entropy} = \sum_{i,j} P(i, j) \log(P(i, j)) \quad (4)$$

4. Homogeneity

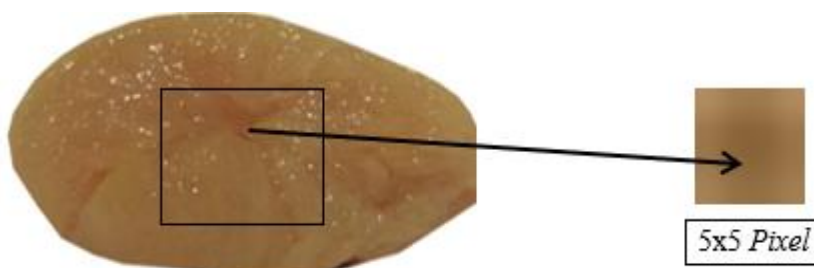
$$\text{Homogeneity} = \sum_{i,j} \frac{p(i, j)}{1 + |i - j|} \quad (5)$$

3. HASIL DAN PEMBAHASAN

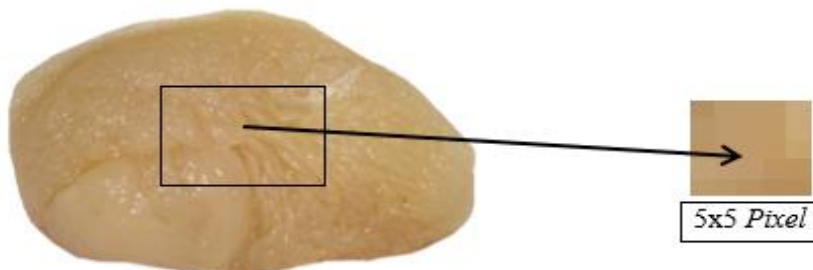
Keakurasian dan kecepatan pendeteksi merupakan hal yang sangat penting dalam pengolahan citra, sehingga metode yang digunakan untuk mendeteksi citra jadi salah satu pertimbangan yang sangat penting agar menghasilkan data yang sesuai dan optimal. Algoritma yang digunakan pada penelitian ini adalah algoritma *Grey Level Co-Occurrence Matrix* dan metode klasifikasi *k-nearest neighbor*. Membuktikan bahwa algoritma GLCM dapat mengekstraksi ciri tekstur dengan baik, GLCM menghasilkan hasil ekstraksi ciri berupa data angka, hasil pengukuran atau disebut data *Continues*, berupa *Contrast*, *Energy*, *Entropy* dan *Homogeneity*. Selain itu pada penelitian yang dilakukan dengan menggunakan metode KNN telah menghasilkan tingkat keakurasian.

Berdasarkan penjelasan di atas, maka dalam penelitian ini akan melakukan analisis, dengan mengkombinasikan algoritma ekstraksi ciri *Grey Level Co-occurrence Matrix* dengan metode klasifikasi *K-Nearest Neighbor* untuk mengklasifikasikan citra dan menguji tingkat keakuratan dalam pengujian dengan berbagai skenario yang telah dipersiapkan. Berdasarkan penjelasan di atas, maka menguraikan sebuah prosedur dalam mengetahui proses pencarian nilai daging ayam busuk dan daging ayam segar sebagai berikut :

1. Menyiapkan citra daging ayam busuk dan daging ayam segar menggunakan kamera *handphone xiami 4X*.
 2. Meresize ukuran citra menjadi 5 x 5 piksel.
 3. Merubah citra RGB menjadi *grayscale* dengan 256 derajat keabuan yang merupakan *default* dari bentuk citra *grayscale*.
 4. Menormalisasikan hasil citra dalam mengubah rentang nilai intensitas *pixel*.
 5. Membuat area tabel matriks
 6. Menentukan piksel ketetangga spasial antara piksel referensi dengan piksel tetangganya, yaitu menentukan nilai sudut (θ).
 7. Menghitung jumlah nilai kookurensi dan mengisikannya pada area kerja yang telah dibuat.
 8. Menormalisasi masing-masing nilai matriks untuk mengubahnya ke bentuk probabilitas.
 9. Melakukan ekstraksi fitur dalam bentuk *Contrast, Energy, Entropy* dan *Homogenity*.
 10. Mengklasifikasikan hasil GLCM dengan metode *K-Nearest Neighbor* serta menentukan parameter *k* (jumlah tetangga terdekat).
 11. Memilih kelas mayoritas serta menghitung nilai akurasi setiap hasil citra yang diklasifikasikan.
- Adapun citra daging ayam segar dan daging ayam busuk sebagai berikut :



Gambar 2. Citra Daging Ayam Busuk Setelah Dilakukan Cropping

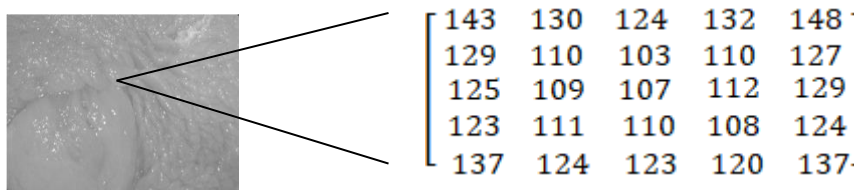


Gambar 3. Citra Daging Ayam Segar Setelah Dilakukan Cropping

3.1 Penerapan Metode *Grey Level Co-Occurrence Matrix* (GLCM)

Metode GLCM ini memiliki empat sudut dan jarak ketetangga satu. Langkah awal yang dilakukan untuk mendapatkan fitur tekstur yaitu melakukan konversi citra RGB menjadi *grayscale*, selanjutnya menentukan ukuran *Co-occurrence Matrix* (CM). Ukuran *Co-occurrence Matrix* ditentukan berdasarkan Persamaan. Matriks *co-occurrence* dibuat dengan menghitung probabilitas kemunculan nilai ketetangaan berdasarkan sudutnya. Setelah mendapatkan *co-occurrence Matrix* selanjutnya yaitu melakukan normalisasi untuk menghilangkan ketergantungan pada ukuran citra. Elemen pada GLCM dinormalisasi untuk mendapatkan jumlah senilai satu. Langkah terakhir merupakan perhitungan fitur statistik GLCM berdasarkan elemen GLCM yang telah di normalisasi.

Metode Matriks *Co-occurrence* merupakan metode untuk ekstraksi ciri, dimana nilai ekstraksi ciri yang akan dicari adalah nilai *Entropy, Contras, Energy* dan *Homogenity*. Sebagai contoh, citra daging ayam I dan daging ayam II dibawah ini telah melalui preprocessing dengan ukuran 5x5 *pixel*.



Gambar 4. Hasil *Grayscale* Daging Ayam 5x5 *pixel*

Misalkan I adalah citra training. Dengan nilai piksel 5x5 seperti matriks di atas untuk menormalisasi nilai citra.

$$I_{i,j} = (I - Min) \frac{newMax - newMin}{Max - Min} + newMin$$

Dimana :

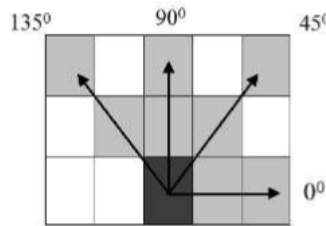
- I = Nilai *Grayscale*
 - Min = Minimal Nilai Warna *Grayscale*
 - Max = Maksimal Nilai Warna *Grayscale*
 - $newMax$ = Nilai Maksimal Baris/Nilai Kolom yang Baru
 - $newMin$ = Nilai Minimal Baris/Nilai Kolom yang Baru
- Menormalisasikan hasil *Grayscale* diatas smpa (1,1.....5,5)

$$I_{1,1}(143 - 103) \frac{12 - 1}{148 - 103} + 1 = 10,778 = 10$$

1. Setelah dinormalisasi masukan hasil perhitungan kedalam matriks sehingga terbentuk matriks seperti dibawah ini :

10	6	6	8	2
7	2	1	7	6
6	2	1	3	7
5	2	2	2	6
9	6	5	5	9

2. Jarak dinyatakan dalam piksel sedangkan untuk orientasi arah dinyatakan dalam derajat dengan standar yaitu 0°, 45°-90° dan 135°. Langkah selanjutnya adalah membuat area kerja matriks dengan ketentuan sudut.



Gambar 5. Orientasi Sudut dan Jarak Pada GLCM

3. Menentukan hubungan antara piksel referensi dan piksel tetangga berdasarkan sudut (θ) yang sudah ditentukan. Salah satu cara untuk merepresentasikan hubungan ini yaitu berupa (1,1), yang menyatakan hubungan dua piksel yang berjajar horizontal dengan piksel bernilai 1. Berdasarkan hubungan komposisi tersebut, jumlah kelompok piksel yang memenuhi hubungan tersebut dihitung. Adapun pembuatan matriks GLCM dapat di lihat pada gambar 3.4 :

10	6	6	8	2
7	2	1	7	6
6	2	1	3	7
5	2	2	2	6
9	6	5	5	9

Pixel	1	2	3	4	5	6	7	8	9	10	11	12
1	0	2	0	0	0	0	1	0	0	0	0	0
2	2	2	0	0	1	2	1	1	0	0	0	0
3	1	0	0	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	0	0	1	1	0	0	1	0	0	0
6	0	2	0	0	1	1	1	1	1	1	0	0
7	1	1	1	0	0	1	0	0	0	0	0	0
8	0	1	0	0	0	1	0	0	0	0	0	0
9	0	0	0	0	1	1	0	0	0	0	0	0
10	0	0	0	0	0	1	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 6. Hubungan Spasial Antara Nilai Piksel

4. Mentranspose *Co-occurrence Matrix*
Matrix Co-occurrence yang sudah didapat selanjutnya di transpose, transpose merupakan proses mengubah matriks baris menjadi kolom, tujuan dari proses ini adalah untuk mendapatkan sudut simetrisnya yaitu sudut 0°. Berikut merupakan hasil pembentukan matriks transpose :

Pixel	1	2	3	4	5	6	7	8	9	10	11	12
1	0	2	0	0	0	0	1	0	0	0	0	0
2	2	2	0	0	1	2	1	1	0	0	0	0
3	1	0	0	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	0	0	1	1	0	0	1	0	0	0
6	0	2	0	0	1	1	1	1	1	0	0	0
7	1	1	1	0	0	1	0	0	0	0	0	0
8	0	1	0	0	0	1	0	0	0	0	0	0
9	0	0	0	0	1	1	0	0	0	0	0	0
10	0	0	0	0	0	1	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0

+

Pixel	1	2	3	4	5	6	7	8	9	10	11	12
1	0	2	1	0	0	0	1	0	0	0	0	0
2	2	2	0	0	1	2	1	1	0	0	0	0
3	0	0	0	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	0	0	1	1	0	0	1	0	0	0
6	0	2	0	0	1	1	1	1	1	0	0	0
7	1	1	1	0	0	1	0	0	0	0	0	0
8	0	1	0	0	0	1	0	0	0	0	0	0
9	0	0	0	0	1	1	0	0	0	0	0	0
10	0	0	0	0	0	1	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0

=

Pixel	1	2	3	4	5	6	7	8	9	10	11	12
1	0	4	1	0	0	0	2	0	0	0	0	0
2	4	4	0	0	2	4	2	2	0	0	0	0
3	1	0	0	0	0	0	2	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	2	0	0	2	2	0	0	2	0	0	0
6	0	4	0	0	2	2	2	2	2	2	0	0
7	2	2	2	0	2	2	0	0	0	0	0	0
8	0	2	0	0	2	2	0	0	0	0	0	0
9	0	0	0	0	2	2	0	0	0	0	0	0
10	0	0	0	0	2	2	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 7. Transpose Matriks

4. Melakukan normalisasi pada setiap matriks ko-okurensi simetris dengan cara menghitung kemungkinan (*probability*) setiap elemen matrik. Cara untuk menghitung kemungkinan pada setiap elemen matrik adalah dengan menggunakan $P(i,j) = (i,j)/\text{total jumlah pasangan}$.
 Dimana total jumlah pasangan = 70

Pixel	1	2	3	4	5	6	7	8	9	10	11	12
1	0	$\frac{4}{70}$	$\frac{1}{70}$	0	0	0	$\frac{2}{70}$	0	0	0	0	0
2	$\frac{4}{70}$	$\frac{4}{70}$	0	0	$\frac{2}{70}$	$\frac{4}{70}$	$\frac{2}{70}$	$\frac{2}{70}$	0	0	0	0
3	$\frac{1}{70}$	0	0	0	0	0	$\frac{2}{70}$	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	$\frac{2}{70}$	0	0	$\frac{2}{70}$	$\frac{2}{70}$	0	0	$\frac{2}{70}$	0	0	0
6	0	$\frac{4}{70}$	0	0	$\frac{2}{70}$	$\frac{2}{70}$	$\frac{2}{70}$	$\frac{2}{70}$	$\frac{2}{70}$	$\frac{2}{70}$	0	0
7	$\frac{2}{70}$	$\frac{2}{70}$	$\frac{2}{70}$	0	0	$\frac{2}{70}$	0	0	0	0	0	0
8	0	$\frac{2}{70}$	0	0	0	$\frac{2}{70}$	0	0	0	0	0	0
9	0	0	0	0	$\frac{2}{70}$	$\frac{2}{70}$	0	0	0	0	0	0
10	0	0	0	0	0	$\frac{2}{70}$	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0

=

Pixel	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0,057143	0,014286	0	0	0	0,028571	0	0	0	0	0
2	0,057143	0,057143	0	0	0,028571	0,057143	0,028571	0,028571	0	0	0	0
3	0,014286	0	0	0	0	0	0,028571	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0,028571	0	0	0,028571	0,028571	0	0	0,028571	0	0	0
6	0	0,057143	0	0	0,028571	0,028571	0,028571	0,028571	0,028571	0,028571	0	0
7	0,028571	0,028571	0,028571	0	0	0,028571	0	0	0	0	0	0
8	0	0,028571	0	0	0	0,028571	0	0	0	0	0	0
9	0	0	0	0	0,028571	0,028571	0	0	0	0	0	0
10	0	0	0	0	0	0,028571	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 8. Tabel Normalisasi matriks dengan probabilitas Daging Ayam Busuk

Melakukan Ekstraksi Fitur Pada Hasil Nilai Daging Ayam Busuk

Langkah terakhir dalam algoritma GLCM ini adalah melakukan ekstraksi fitur, ekstraksi fitur dilakukan pada semua data. Baik citra *training* maupun *testing*. Pada penelitian ini penulis menggunakan empat ekstraksi fitur yaitu *Contrast*, *Energy*, *Entropy* dan *Homogeneity*. Ekstraksi fitur dihitung dengan rumus sebagai berikut:

a. *Contrast*

$$Contrast = \sum_{i,j} P(i,j) |P_d(i-j)|^2$$

Dimana :

i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatan, sedangkan p (i,j) adalah Probabilitas kolom(i,j).

Syarat : Ketika nilai i dan j sama, sel berada pada diagonal dan (i-j) = 0. Nilai-nilai ini merepresentasikan *pixel* yang keseluruhannya mirip dengan tetangga mereka, sehingga mereka diberi bobot 0.

Contoh perhitungan :

$$P(i,j) |P_d(i-j)|^2$$

Dimana P adalah *pixel*, sedangkan i adalah kolom dan j adalah baris.

$$P(5,5).P(5-5)^2 = (0,057143).(1-2)^2 = 0,057143.1 = 0,057143$$

Berikut tabel hasil perhitungan *Contrast* :

Tabel 3. Hasil Perhitungan *Contrast*

0	0,05714	0,0571	0	0	0	0,0286	0	0	0	0	0
0,0571	0,0571	0	0	0,2571	0,9143	0,7143	1,0286	0	0	0	0
0,0571	0	0	0	0	0	0,4571	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0,2572	0	0	0	0,0286	0	0	0,4571	0	0	0
0	0,9143	0	0	0,0286	0	0,0286	0,1143	0,2571	0,4571	0	0
1,0286	0,7143	0,4571	0	0	0,0286	0	0	0	0	0	0
0	1,0286	0	0	0	0,1143	0	0	0	0	0	0
0	0	0	0	0,4571	0,2571	0	0	0	0	0	0
0	0	0	0	0	0,4571	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Berdasarkan hasil perhitungan *contras* di atas, maka menjumlahkan semua nilai-nilai setiap baris dan kolom, dengan rumus:

Total nilai *contrast* = (i+j), dimana i adalah baris sedangkan j adalah kolom.

$$\text{Total nilai } contrast = 10,77124$$

b. *Energy*

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

Dimana:

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

$$\text{Contoh perhitungan } Energy P(i,j)^2 = (0,057143)^2 = 0,00327$$

Tabel 4. Hasil Perhitungan *Energy*

0	0,00327	0,0002	0	0	0	0,00082	0	0	0	0	0
0,00327	0,00327	0	0	0,00082	0,00327	0,00082	0,00082	0	0	0	0
0,0002	0	0	0	0	0	0,00082	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0,00082	0	0	0,00082	0,00082	0	0	0,00082	0	0	0
0	0,00327	0	0	0,00082	0,00082	0,00082	0,00082	0,00082	0,00082	0	0
0,00082	0,00082	0,00082	0	0	0,00082	0	0	0	0	0	0
0	0,00082	0	0	0	0,00082	0	0	0	0	0	0
0	0	0	0	0,00082	0,00082	0	0	0	0	0	0
0	0	0	0	0	0,00082	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Berdasarkan hasil perhitungan *energy* di atas, maka menjumlahkan semua nilai-nilai setiap baris dan kolom, dengan rumus:

Total nilai *energy* = (i+j), dimana i adalah baris sedangkan j adalah kolom.

Total nilai *energy* = **0,02824**

c. *Entropy*

$$\text{Rumus : Entropy} = - \sum_{i,j} P(i,j) \log(P(i,j))$$

Dimana:

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

$$\text{Perhitungan Entropy} = -P(i,j) \log(P(i,j)) = -P(0,057143) \cdot \log(P(0,057143))$$

Tabel 5. Hasil Perhitungan *Entropy*

0	0,071031	0,026359	0	0	0	0,044116	0	0	0	0	0
0,071031	0,071031	0	0	0,044116	0,071031	0,044116	0,044116	0	0	0	0
0,026359	0	0	0	0	0	0,044116	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0,044116	0	0	0,044116	0,044116	0	0	0,044116	0	0	0
0	0,071031	0	0	0,044116	0,044116	0,044116	0,044116	0,044116	0,044116	0	0
0,044116	0,044116	0,044116	0	0	0,044116	0	0	0	0	0	0
0	0,044116	0	0	0	0,044116	0	0	0	0	0	0
0	0	0	0	0,044116	0,044116	0	0	0	0	0	0
0	0	0	0	0	0,044116	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Berdasarkan hasil perhitungan *entropy* di atas, maka menjumlahkan semua nilai-nilai setiap baris dan kolom, dengan rumus:

Total nilai *entropy* = (i+j), dimana i adalah baris sedangkan j adalah kolom.

Total nilai *entropy* = **1,466657**

b. *Homogeneity*

$$\text{Homogeneity} = \sum_{i,j} \frac{p(i,j)}{1 + |i - j|}$$

Contoh menghitung *homogeneity* : $p(i,j)/1+|i-j| = p(0,057143)/1+|1-2| =$

Tabel 6. Perhitungan *Homogeneity*

0	0,057143	0,014286	0	0	0	0,00571	0	0	0	0	0
0,028572	0,057143	0	0	0,014286	0,019048	0,007143	0,005714	0	0	0	0
0,004762	0	0	0	0	0	0,009552	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0,007143	0	0	0,028571	0,028571	0	0	0,009552	0	0	0
0	0,011429	0	0	0,014286	0,028571	0,028571	0,028571	0,014286	0,009552	0	0
0,004082	0,004762	0,005714	0	0	0,014286	0	0	0	0	0	0
0	0,004082	0	0	0	0,009524	0	0	0	0	0	0
0	0	0	0	0,005714	0,007143	0	0	0	0	0	0
0	0	0	0	0	0,005714	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Berdasarkan hasil perhitungan *homogeneity* di atas, maka menjumlahkan semua nilai-nilai setiap baris dan kolom, dengan rumus :

Total nilai *homogeneity* = (i+j), dimana i adalah baris sedangkan j adalah kolom.

Total nilai *homogeneity* = **0,493483**

3.2 Melakukan Klasifikasi Dengan Metode K-Nearest Neighbor (K-NN)

Untuk itu setelah dilakukan ekstraksi fitur dengan menggunakan fitur ekstraksi GLCM selanjutnya citra harus diklasifikasi untuk bisa membedakan kedua jenis citra tersebut. Dalam penelitian ini penulis menggunakan K-NN yaitu metode klasifikasi berdasarkan data pembelajaran yang jaraknya dekat dengan tetangga terdekat (k). Untuk perhitungan jaraknya penulis menggunakan jarak *Euclidean*. Proses klasifikasi dilakukan dengan langkah berikut ini : Misal diketahui 4 buah citra, citra

ini termasuk kedalam citra training dan menambah satu citra untuk diuji (citra 5). Pada penelitian ini ekstraksi fitur yang digunakan adalah 4 ekstraksi fitur diantaranya *contrast*, *energy*, *entropy* dan *homogeneity*. Misalnya didapatkan nilai dari hasil perhitungan dari fitur ekstraksi GLCM adalah sebagai berikut :

Tabel 7. Nilai Hasil Ekstrasi Fitur GLCM

Citra	Contrast	Energy	Entropy	Homogeneity
1	10,7712	0,02824	1,466657	0,493483
2	7,3326	0,14788	1,455577	0,61336
3	12,138	0,141518	1,125066	0,580115

Sebagai contoh citra 1 sampai dengan citra 2 merupakan citra training. Dan citra 3 adalah citra testing yang akan diklasifikasikan. Citra testing ini merupakan citra pengujian dalam mencari jarak *euclidean* antara citra lainnya. Langkah pertama yang harus dihitung adalah jarak *Euclidean*.

1. Menentukan parameter nilai k , atau tetangga terdekatnya. Misal $k = 1$.
2. Menhitung jarak Euclidean citra dengan semua citra *training*.
3. Menghitung jarak jarak *Euclidean*.

Jarak *Euclidean* antara citra 1 dan 3

$$= \sqrt{\sum_{i=1}^n ((10,7712 - 12,138)^2 + (0,02824 - 0,141518)^2 + (1,466657 - 1,125066)^2)}$$

$$\sqrt{\sum_{i=1}^n ((0,493483 - 0,580115)^2)} = 2,005164$$

Jarak *Euclidean* antara citra 2 dan 3

$$= \sqrt{\sum_{i=1}^n ((7,3326 - 12,138)^2 + (0,14788 - 0,141518)^2 + (1,455577 - 1,125066)^2)}$$

$$\sqrt{\sum_{i=1}^n ((0,61336 - 0,580115)^2)} = 23,202252$$

4. Mengurutkan hasil perhitungan jarak antar citra berdasarkan jarak k minimum. Menentukan ranking jarak antar citra berdasarkan k minimum untuk menentukan nilai terdekat pada jarak $k=1$ dapat dilihat pada tabel 3.20.

Tabel 8. Urutan jarak citra *training* dengan citra *testing*.

Jarak citra dengan citra 1	Ranking jarak	Tetangga $k=1$	Daging Ayam
Citra 1	2,005164	Ya	Busuk
Citra 2	23,202252	Tidak	Segar

Berdasarkan hasil fitur dengan nilai yang terdekat pada tabel di atas dalam menentukan fitur yang telah mendekati nilai $k=1$ yaitu termasuk citra daging ayam busuk, maka data baru yang menjadi perbandingan termasuk kelas daging ayam busuk.



3.3 Hasil Pengujian




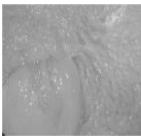

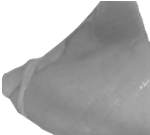




Berdasarkan dari *form* pengujian program untuk mengetahui daging ayam busuk dan daging ayam segar pada aplikasi bahasa pemrograman PHP dapat disajikan dibawah ini : dapat dilihat pada tabel 9 dibawah ini :

1. Nilai Hasil Ekstrasi Fitur GLCM

Nilai hasil ekstrasi fitur GLCM merupakan nilai hasil dari citra RGB yang diinput dan mengubahnya ke *grayscale* yang disimpan kedalam *database*. Hasil ekstrasi fitur GLCM dapat dilihat pada tabel 4.1 dibawah ini :

Tabel 9. Nilai Hasil Ekstrasi Fitur GLCM

Citra RGB	Grayscale	Contrast	Energy	Entropy	Homogeneity	Daging Ayam
		10,7712	0,02824	1,466657	0,493483	Busuk

Citra RGB	Grayscale	Contrast	Energy	Entropy	Homogeneity	Daging Ayam
		4,6902	0,097334	1,141942	0,719724	Busuk
		7,3326	0,14788	1,455577	0,61336	Segar
		3,1668	0,108676	1,049858	0,750614	Segar
		12,138	0,141518	1,125066	0,580115	?
		7,3326	0,14788	1,455577	0,61336	Segar

2. Hasil Euclidean Serta Mengurutkan Jarak

Hasil euclidean merupakan hasil perhitungan nilai jarak $k=3$ serta menentukan jarak antara nilai citra training dan citra testing, sehingga menentukan akurasi setiap citra. Hasil euclidean ini dapat dilihat pada tabel 10 dibawah ini :

Tabel 10. Nilai Akurasi Dengan Jarak $k=3$


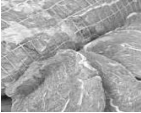

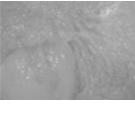



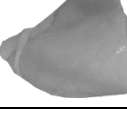
Euclidean	Ranking jarak	Tetangga $k=1$	Akurasi	Daging Ayam
2,005164	4	Tidak	40%	Busuk
7,449259	3	Ya	49%	Busuk
23,202252	1	Ya	64%	Segar
8,973195	2	Ya	79%	Segar

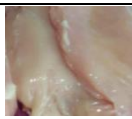

Berdasarkan hasil pengujian menghasilkan fitur dengan nilai yang terdekat pada tabel di atas dalam menentukan fitur yang telah mendekati nilai $k=3$ yaitu termasuk citra daging ayam segar, maka data baru yang menjadi perbandingan termasuk kelas daging ayam segar.

3. Nilai Hasil Citra Data Testing

Nilai hasil citra data testing merupakan salah satu hasil citra testing yang menentukan data testing daging ayam berdasarkan nilai mayoritas dengan nilai tetangga terdekat $k=3$. Adapun tabel hasil citra testing dibawah ini :

Tabel 11. Hasil Citra Testing

Citra RGB	Grayscale	Contrast	Energy	Entropy	Homogeneity	Daging Ayam
		10,7712	0,02824	1,466657	0,493483	Busuk
		4,6902	0,097334	1,141942	0,719724	Busuk
		7,3326	0,14788	1,455577	0,61336	Segar
		3,1668	0,108676	1,049858	0,750614	Segar

Citra RGB	Grayscale	Contrast	Energy	Entropy	Homogeneity	Daging Ayam
		12,138	0,141518	1,125066	0,580115	Segar

Pengujian proses citra training dan citra testing, dimana hasil citra testing yaitu terdapat pada mayoritas daging ayam segar. Dengan hasil urutan terkecil diambil dari nilai $k=3$, maka perbandingannya adalah antara citra training dan citra testing. Maka dapat disimpulkan daging ayam dengan nilai dominan terdapat pada citra daging ayam segar.

4. KESIMPULAN

Adapun kesimpulan dari hasil penelitian dalam menentukan citra berekstensi JPG yang diklasifikasikan yaitu daging ayam busuk dan daging ayam segar sebagai citra training dan citra testing. Dengan menggunakan fitur ekstraksi GLCM dengan empat fitur ekstraksi yaitu Contrast, Energy, Entropy dan Homogeneity serta metode klasifikasi K-NN mampu membedakan citra daging ayam busuk dan daging ayam segar. Hasil akurasi yang diperoleh dari penelitian ini mencapai 79% yaitu nilai akurasi citra daging ayam segar dengan menggunakan sudut pengenalan pada GLCM 0° dan nilai k pada K-NN bernilai $k=3$. Sedangkan akurasi yang paling terendah adalah sebesar 40% yaitu nilai akurasi citra daging ayam busuk pada saat GLCM menggunakan sudut 0° dan nilai k pada K-NN sebesar 3. Hal ini menunjukkan perubahan pada sudut GLCM dan besarnya nilai k pada K-NN akan mempengaruhi besarnya akurasi yang didapatkan.

REFERENCES

- [1] Satrio A. Wibowo, Bambang Hidayat, Unang Sunarya, "Simulasi dan Analisis Pengenalan Citra Daging Sapi dan Babi Dengan Metode GLCM", Seminar Nasional Inovasi dan Aplikasi, Teknologi Di Industri (SENIATI) 2016, ISSN 2085-4218.
- [2] O. N. Shpakov and G. V. Bogomolov, "Technogenic activity of man and local sources of environmental pollution," Studies in Environmental Science, vol. 17, no. C, pp. 329–332, 1981.
- [3] H. Leidiyana, "Penerapan Algoritma K-Nearest Neighbor Untuk Penentuan Resiko Kredit Kepemilikan Kendaraan Bermotor," Jurnal Penelitian Ilmu Komputer, System Embedded & Logic, vol. 1, no. 1, pp. 65–76, 2013.
- [4] F. M. Amin, "Identifikasi Citra Daging Ayam Berformalin Menggunakan Metode Fitur Tekstur dan K-Nearest Neighbor (K-NN)," Jurnal Matematika "MANTIK," vol. 4, no. 1, pp. 68–74, 2018.
- [5] M. Lestari, "Penerapan Algoritma Klasifikasi Nearest Neighbor (K-NN) Untuk Mendeteksi Penyakit Jantung," Faktor Exacta, vol. 7, no. September 2010, pp. 366–371, 2014.
- [6] A. Nugraha, pratama dwi., Said al faraby, "Klasifikasi Dokumen Menggunakan Metode Knn Dengan Information Gain," eProceedings of Engineering, vol. 5, no. 1, pp. 1541–1550, 2018.
- [7] F. A. Susanto, "Identifikasi Daging Sapi Dan Daging Babi Menggunakan Fitur Ekstraksi Grey Level Co-Occurrence Matrix Dan K-Nearest Neighbor," Skripsi, Fakultas Ilmu Komputer, pp. 1–8, 2015.