



Support Vector Machine Untuk Klasifikasi Citra Jenis Daging Berdasarkan Tekstur Menggunakan Ekstraksi Ciri *Gray Level Co-Occurrence Matrices* (GLCM)

Neneng^{a*}, Kusworo Adi^b, R. Rizal Isnanto^c

^a STMIK Teknokrat Bandar Lampung

^b Fakultas Sains dan Matematika Universitas Diponegoro

^c Fakultas Teknik Universitas Diponegoro

Naskah Diterima : 14 Januari 2016; Diterima Publikasi : 15 Maret 2016

DOI: 10.21456/vol6iss1pp1-10

Abstract

Texture is one of the most important features for image analysis, which provides informations such as the composition of texture on the surface structure, changes of the intensity, or brightness. Gray level co-occurrence matrix (GLCM) is a method that can be used for statistical texture analysis. GLCM has proven to be the most powerful texture descriptors used in image analysis. This study uses the four-way GLCM 0° , 45° , 90° , and 135° . Support vector machine (SVM) is a machine learning that can be used for image classification. SVM has a high generalization capability without any requirement of additional knowledge, even with the high dimension of the input space. The data used in this study are the image of goat meat, buffalo meat, horse meat, and beef with shooting distance 20 cm, 30 cm and 40 cm. The result of this study shows that the best recognition rate of 87.5% was taken at a distance of 20 cm with neighboring pixels distance $d = 2$ in the direction GLCM 135° .

Keywords : Image; GLCM; Tecture; SVM

Abstrak

Tekstur merupakan salah satu ciri yang paling penting untuk analisis citra, dimana tekstur menyediakan informasi mengenai susunan stuktur pada permukaan, perubahan intensitas, atau kecerahan warna. *Grey level co-occurrence matrix* (GLCM) merupakan metode yang dapat digunakan untuk analisis tekstur statistik. GLCM telah terbukti menjadi deskriptor tekstur yang paling kuat yang digunakan dalam analisis citra. Penelitian ini menggunakan empat arah GLCM yakni 0° , 45° , 90° , dan 135° . *Support vector machine* (SVM) merupakan mesin pembelajaran yang dapat digunakan untuk klasifikasi citra. SVM memiliki kemampuan generalisasi yang tinggi tanpa persyaratan pengetahuan tambahan, bahkan dengan dimensi yang tinggi dari ruang input. Data yang digunakan dalam penelitian ini adalah citra daging kambing, daging kerbau, daging kuda, dan daging sapi dengan jarak pengambilan 20 cm, 30 cm, dan 40 cm. Penelitian ini menghasilkan tingkat pengenalan terbaik yakni 87,5% berada pada jarak pengambilan 20 cm dengan jarak piksel tetangga $d=2$ pada arah GLCM 135° .

Kata kunci : Citra; GLCM; Tekstur; SVM

1. Pendahuluan

Tekstur merupakan pola atau karakteristik yang teratur maupun acak pada sebuah objek atau gambar (Schwartz dan Pedrini, 2006). Tekstur adalah salah satu ciri yang paling penting untuk analisis citra, dimana tekstur menyediakan informasi mengenai susunan stuktur pada permukaan, perubahan intensitas, atau kecerahan warna (Siqueira *et al.*, 2013). Ciri tekstur dapat digolongkan menjadi empat jenis, yakni tekstur statistik, tekstur struktur, tekstur berbasis model, dan tekstur berbasis transformasi.

Diantara empat jenis tekstur tersebut, tekstur statistik yang paling banyak digunakan dalam industri makanan untuk melakukan penilaian kualitas maupun klasifikasi (Bharati *et al.*, 2004).

Metode yang dapat digunakan untuk analisis tekstur statistik diantaranya adalah *grey level co-occurrence matrix* (GLCM), *grey level pixel-run length matrix*, dan *neighbouring grey level dependence matrix* (Zheng *et al.*, 2006). Diantara metode tersebut, GLCM yang paling banyak digunakan di mana ciri tekstur diekstraksi dengan beberapa pendekatan statistik dari matriks *co-*

*) Penulis korespondensi: neneng060880@yahoo.co.id

occurrence (Haralick *et al.*, 1973). GLCM telah terbukti menjadi deskriptor tekstur yang paling kuat yang digunakan dalam analisis citra (Siqueira *et al.*, 2013).

Dalam proses analisis tekstur, perhitungan ciri dibutuhkan untuk setiap tekstur dimana ciri-ciri ini mengandung informasi karakteristik yang dapat di gambarkan secara visual maupun tidak (Basset *et al.*, 2000). Seleksi ciri dalam klasifikasi memiliki tiga tujuan yaitu: (1) untuk mengurangi biaya penggalian ciri, (2) untuk meningkatkan akurasi klasifikasi, dan (3) untuk meningkatkan keandalan estimasi kinerja (Kudo dan Sklansky, 2000).

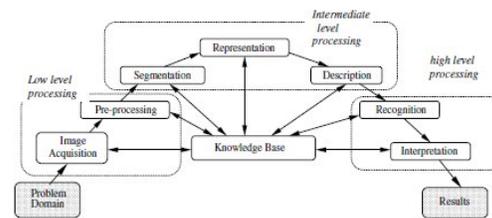
Penelitian ilmiah tentang tekstur masih terbatas karena keragaman pola tekstur, sehingga penelitian tentang tekstur gambar menjadi lebih sulit dan sejauh ini belum ada metode karakterisasi tekstur ideal yang tersedia (Patel, Davies, dan Hannah, 1996). Namun demikian, beberapa ide tentang bagaimana tekstur dapat dikembangkan dibagi dalam dua aspek yang berbeda. Pertama, tekstur umumnya disebut sebagai ketergantungan antara piksel dan piksel tetangga dalam gambar. Kedua, tekstur didefinisikan sebagai variasi dari nilai intensitas di seluruh piksel (Haralick, 1979).

Terdapat beberapa mesin pembelajaran untuk klasifikasi citra, seperti jaringan syaraf tiruan, logika fuzzy, algoritma genetika, dan *support vector machine* (SVM). Struktur dari jaringan syaraf tiruan sulit untuk memahami dan menangkap atribut yang unik dalam tahap pelatihan. Logika fuzzy memerlukan pengetahuan sebelumnya tentang sistem, sedangkan algoritma genetika memiliki komplikasi pada tahap pelatihan atau output data (Seetha *et al.*, 2008). Dari berbagai mesin pembelajaran yang tersedia tersebut, SVM memiliki kemampuan generalisasi yang tinggi tanpa persyaratan pengetahuan tambahan, bahkan dengan dimensi yang tinggi dari ruang input (Burgess, 1998). SVM merupakan teknik yang sangat berguna untuk klasifikasi data dan masalah regresi yang diciptakan oleh Vladimir Vapnik (Salat dan Osowski, 2004).

2. Kerangka Teori

2.1. Pengolahan Citra dan Analisis Citra

Pengolahan citra dan analisis citra merupakan inti dari visi komputer (Krutz *et al.*, 2000). Pengolahan citra/analisis mencakup serangkaian langkah yang secara garis besar dibagi menjadi tiga tingkatan yaitu pengolahan tingkat rendah, pengolahan tingkat menengah, dan pengolahan tingkat tinggi yang ditunjukkan pada Gambar 1 (Sun, 2000).



Gambar 1. Tingkatan Proses Pengolahan Citra (Sun, 2000)

2.2. Jenis Citra

Dalam pemrosesan citra, terdapat tiga jenis citra yang umum digunakan yakni citra berwarna, citra berskala keabuan, dan citra biner yang dijelaskan sebagai berikut (Kadir dan Susanto, 2013):

1. Citra berwarna

Citra berwarna atau yang sering disebut citra RGB adalah jenis citra yang menyajikan warna dalam bentuk komponen R (merah), G (hijau), dan B (biru). Setiap komponen warna tersebut menggunakan delapan bit yang nilainya berkisar antara 0 sampai dengan 255. Dengan demikian, komponen warna yang dapat disajikan mencapai 16.581.375 warna. Tabel 1 berikut menjelaskan contoh warna dan nilai RGB-nya.

Tabel 1. Warna dan nilai penyusun warna

Warna	R	G	B
Merah	255	0	0
Hijau	0	255	0
Biru	0	0	255
Hitam	0	0	0
Putih	255	255	255
Kuning	0	255	255

2. Citra berskala keabuan

Citra jenis ini menangani gradasi warna hitam dan putih yang menghasilkan efek warna abu-abu. Pada jenis gambar ini, warna dinyatakan dengan intensitas yang berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam, sedangkan nilai 255 menyatakan putih.

3. Citra biner

Citra biner merupakan citra yang setiap pikselnya hanya dinyatakan dengan nilai dari dua kemungkinan yakni nilai 0 dan 1. Nilai 0 menyatakan warna hitam, sedangkan nilai 1 menyatakan warna putih.

Dalam prakteknya, mengingat banyak pemrosesan yang bekerja pada skala keabuan, maka citra berwarna seringkali harus dikonversi terlebih dahulu ke dalam bentuk citra berskala keabuan. Untuk mengubah citra RGB menjadi citra skala keabuan dapat dilakukan dengan rumus sebagai berikut:

$$\text{Skala keabuan} = 0.299R + 0.587G + 0.144B \quad (1)$$

$$\text{Skala keabuan} = 0.333R + 0.333G + 0.333B \quad (2)$$

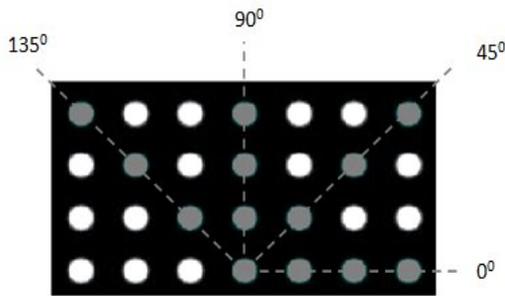
2.3. GLCM

Kulkarni (1994) dalam Kadir dan Susanto (2013) mengatakan bahwa GLCM pertama kali diusulkan oleh Haralick pada tahun 1973 dengan 28 fitur untuk menjelaskan pola spasial. Hall-Bayer (2007) dalam Kadir dan Susanto (2013) menjelaskan GLCM menggunakan perhitungan tekstur pada orde kedua. Pada orde pertama, pengukuran tekstur menggunakan perhitungan statistik didasarkan pada nilai piksel citra semata, seperti varians, dan tidak memperhatikan hubungan ketetanggaan piksel. Sedangkan hubungan antar pasangan dua piksel citra asli diperhitungkan pada orde kedua.

Misalkan, $f(x, y)$ adalah citra dengan ukuran N_x dan N_y yang memiliki piksel dengan kemungkinan hingga L level dan \vec{r} adalah vektor arah offset spasial. $GLCM_{\vec{r}}(i, j)$ didefinisikan sebagai jumlah piksel dengan $j \in 1, \dots, L$ yang terjadi pada offset \vec{r} terhadap piksel dengan nilai $i \in 1, \dots, L$, yang dapat dinyatakan dalam rumus sebagai berikut (Newsam dan Kammath, 2005 dalam Kadir dan Susanto, 2012).

$$GLCM_{\vec{r}}(i, j) = \frac{\#\{(x_1, y_1), (x_2, y_2) \in (N_x, N_y) \times (N_x, N_y) \mid f(x_1, y_1) = i, f(x_2, y_2) = j, (x_2 - x_1, y_2 - y_1) = \vec{r}\}}{(N_x - x_1)(N_y - y_1)} \quad (3)$$

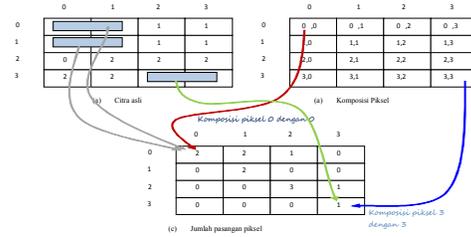
Dalam hal ini, offset \vec{r} dapat berupa sudut dan/atau jarak. Gambar 2 berikut memperlihatkan empat arah untuk GLCM (Kadir dan Susanto, 2012).



Gambar 2. Contoh arah untuk GLCM dengan sudut 0°, 45°, 90°, dan 135° (Kadir dan Susanto, 2012)

Untuk kepentingan ilustrasi, ketetanggaan piksel dapat dipilih ke arah timur (kanan). Salah satu cara untuk merepresentasikan hubungan ini yaitu berupa (1,0), yang menyatakan hubungan dua piksel yang berjajar horizontal dengan piksel bernilai 1 diikuti dengan piksel bernilai 0. Berdasarkan komposisi tersebut, jumlah kelompok piksel yang memenuhi hubungan tersebut dihitung. Hal ini diilustrasikan pada Gambar 3 sebagai berikut (Kadir dan Susanto, 2012).

Gambar 3. Penentuan awal matriks GLCM berbasis pasangan dua piksel (Kadir dan Susanto, 2012)



Gambar 3. Penentuan awal matriks GLCM berbasis pasangan dua piksel (Kadir dan Susanto, 2012)

Matriks pada Gambar 2.3 disebut dengan matriks *framework*. Matriks ini perlu diolah menjadi matriks yang simetris dengan cara menambahkan dengan hasil transposnya, sebagaimana diperlihatkan pada Gambar 4 (Kadir dan Susanto, 2012).

$$\begin{bmatrix} 2 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

Transpos

GLCM sebelum dinormalisasi

Gambar 4. Contoh pembentukan matriks GLCM yang simetris (Kadir dan Susanto, 2012)

Nilai-nilai elemen GLCM perlu dinormalisasi sehingga jumlahnya bernilai 1 untuk menghilangkan ketergantungan pada ukuran citra. Dengan demikian, contoh di atas menjadi seperti berikut (Kadir dan Susanto, 2012).

$$\begin{bmatrix} 4 & 2 & 1 & 0 \\ 24 & 24 & 24 & 24 \\ 2 & 4 & 0 & 0 \\ 24 & 24 & 24 & 24 \\ 1 & 0 & 6 & 1 \\ 24 & 24 & 24 & 24 \\ 0 & 0 & 1 & 2 \\ 24 & 24 & 24 & 24 \end{bmatrix}$$

Hanya beberapa besaran yang diusulkan Haralick yang dipakai untuk mendapatkan fitur GLCM. Sebagai contoh, Newsam dan Kammath (2005) hanya menggunakan lima besaran untuk GLCM yakni *angular second moment* (ASM), *contrast*, *inverse different moment* (IDM), entropi, dan korelasi (Kadir dan Susanto, 2013).

ASM merupakan ukuran homogenitas citra dihitung dengan cara sebagai berikut:

$$ASM = \sum_{i=1}^L \sum_{j=1}^L (GLCM(i,j))^2 \tag{4}$$

Dalam hal ini, L menyatakan jumlah level yang digunakan untuk komputasi.

Kontras merupakan ukuran keberadaan variasi atas keabuan piksel citra, dihitung dengan cara sebagai berikut:

$$Kontras = \sum_{i=1}^L \sum_{j=1}^L \left\{ \sum_{t=1}^L GLCM(i,j) \right\} \tag{5}$$

IDM digunakan untuk mengukur homogenitas, dihitung dengan cara sebagai berikut:

$$IDM = \sum_{i=1}^L \sum_{j=1}^L \frac{(GLCM(i,j))^2}{1 + (i-j)^2} \tag{6}$$

Entropi menyatakan ukuran ketidakteraturan aras keabuan di dalam citra. Nilainya tinggi jika elemen-elemen GLCM mempunyai nilai yang relatif sama. Nilai rendah jika elemen-elemen GLCM dekat dengan nilai 0 atau 1. Rumus untuk menghitung entropi adalah sebagai berikut:

$$Entropi = \sum_{i=1}^L \sum_{j=1}^L (GLCM(i,j) \log (GLCM(i,j))) \tag{7}$$

Korelasi yang merupakan ukuran ketergantungan linear antar nilai aras keabuan dalam citra dihitung menggunakan rumus sebagai berikut:

$$Korelasi = \sum_{i=1}^L \sum_{j=1}^L \frac{(ij)(GLCM(i,j)\mu_i\mu_j)}{\sigma_i\sigma_j} \tag{8}$$

dengan

$$\mu_i = \sum_{i=1}^L \sum_{j=1}^L i \cdot GLCM(i,j) \tag{9}$$

$$\mu_j = \sum_{i=1}^L \sum_{j=1}^L j \cdot GLCM(i,j) \tag{10}$$

$$\sigma_i^2 = \sum_{i=1}^L \sum_{j=1}^L GLCM(i,j) (i - \mu_i)^2 \tag{11}$$

$$\sigma_j^2 = \sum_{i=1}^L \sum_{j=1}^L GLCM(i,j) (j - \mu_j)^2 \tag{12}$$

2.4. Support Vector Machine

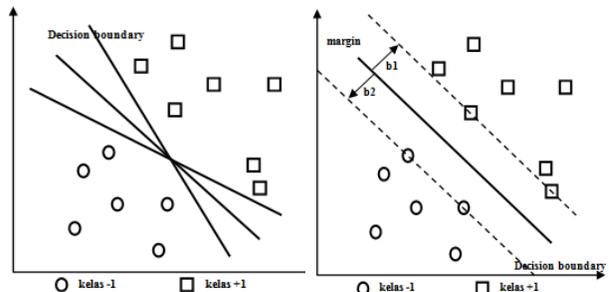
SVM merupakan sistem pembelajaran yang menggunakan ruang hipotesis berupa fungsi-fungsi linier dalam sebuah ruang ciri (*feature space*) berdimensi tinggi, dilatih dengan algoritma pembelajaran yang didasarkan pada teori optimasi

dengan mengimplementasikan bias pembelajaran yang berasal dari teori pembelajaran statistik (Nello dan Taylor, 2000).

SVM merupakan metode klasifikasi yang kini banyak dikembangkan dan diterapkan. Metode ini berasal dari teori pembelajaran statistik yang menjanjikan dan memberikan hasil yang lebih baik dibandingkan dengan metode yang lainnya. SVM bekerja sangat baik pada himpunan data berdimensi tinggi. SVM yang menggunakan teknik kernel harus memetakan data asli dari dimensi asalnya menjadi dimensi lain yang relatif lebih tinggi. Pada metode ANN, selama proses pelatihan semua data latih akan dipelajari. Berbeda dengan SVM yang hanya sejumlah data terpilih sajalah yang digunakan dalam klasifikasi yang akan dipelajari. Berbeda juga dengan Nearest Neighbor yang menyimpan semua data latih yang akan digunakan pada saat prediksi, SVM hanya menyimpan sebagian kecil data latih untuk digunakan pada saat prediksi. Hal inilah yang menjadi kelebihan SVM karena tidak semua data latih dilibatkan pada saat pelatihan (Prasetyo, 2014)

2.6. Konsep SVM

Ide dasar dari SVM adalah memaksimalkan batas *hyperplane* yang dijelaskan pada Gambar 5.



Gambar 5. Batas keputusan yang mungkin untuk set data (Prasetyo, 2014)

Nugroho (2007) dalam Eko Prasetyo (2014) menjelaskan konsep klasifikasi dengan SVM adalah sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas data pada *input space*. Pada gambar 2.5 memperlihatkan beberapa data yang merupakan anggota dari dua buah kelas data yakni +1 dan -1. Bentuk lingkaran merupakan simbol data yang tergabung dalam kelas -1, sedangkan bentuk bujur sangkar merupakan simbol data yang tergabung dalam kelas +1. *Hyperplane* pemisah terbaik antara kedua kelas tersebut dapat ditemukan dengan mengukur margin *hyperplane* dan mencari titik maksimalnya. Margin merupakan jarak antara *hyperplane* tersebut dengan data terdekat dari masing-masing kelas. Data yang paling dekat inilah yang disebut dengan *support vector*. Pada gambar 2 b di atas menunjukkan *hyperplane* terbaik karena terletak di tengah-tengah kedua kelas. Data lingkaran dan bujur sangkar yang

dilewati garis putus-putus (garis batas margin) adalah *support vector*. Inti dari proses pelatihan pada SVM ini adalah usaha untuk mencari lokasi *hyperplane* (Prasetyo, 2014).

2.7. *Hyperplane SVM*

Secara matematika, formulasi problem optimasi SVM untuk kasus klasifikasi linear di dalam *primal space* adalah sebagai berikut (Prasetyo, 2012):

$$\min \frac{1}{2} \|w\|^2$$

Syarat $y_i(w \cdot x_i + b) \geq 1, i = 1, \dots, l$ (13)

Optimalisasi ini dapat diselesaikan dengan pengali Langrange:

$$L_p = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i y_i (w \cdot x_i + b) - 1$$
 (14)

α_i adalah Langrange Multiplier yang berkorespondensi dengan x_i . Nilai α_i adalah nol atau positif. Untuk meminimalkan pengali Langrange, persamaan 14 harus diturunkan pada w dan b , dan diset dengan nilai nol untuk syarat optimalisasi di atas.

Syarat 1: $\frac{\partial L_p}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^N \alpha_i y_i x_i$ (15)

Syarat 2: $\frac{\partial L_p}{\partial b} = 0 \Rightarrow w = \sum_{i=1}^N \alpha_i y_i = 0$ (16)

N adalah jumlah data yang menjadi support vector. Karena Langrange Multiplier (α) tidak diketahui nilainya, persamaan di atas tidak dapat diselesaikan secara langsung untuk mendapatkan w dan b .

Untuk menyelesaikan masalah tersebut, persamaan 14 dimodifikasi menjadi kasus pemaksimalan, syarat optimal untuk dualitasnya menggunakan kendala (*constraint*) Karush Kuhn Tucker (KKT) berikut:

Syarat 1: $\alpha_i [y_i (w \cdot x_i + b) - 1] = 0$ (17)

Syarat 2: $\alpha_i \geq 0, i = 1, 2, \dots, N$ (18)

Dengan menerapkan kendala pada persamaan 17 dan 18, dipastikan bahwa nilai Langrange Multiplier sama banyaknya dengan data latih, tetapi sebenarnya banyak dari data latih yang Langrange Multiplier-

nya sama dengan nol (karena hanya beberapa saja yang akan menjadi support vector) ketika menerapkan syarat pertama. Kendala di atas menyatakan bahwa Langrange α_i harus nol, kecuali untuk data latih x_i yang memenuhi persamaan

$$y_i(w \cdot x_i + b) = 1$$
 (19)

Data latih itu dengan $\alpha_i >$, terletak pada *hyperplane* b_{i1} atau b_{i2} dan disebut support vector. Data latih yang tidak terletak di *hyperplane* tersebut mempunyai $\alpha_i = 0$. Persamaan 17 dan 18 juga menyarankan parameter w dan b , yang mendefinisikan *hyperplane*, hanya tergantung pada support vector. Walaupun bias diselesaikan, masalah optimalisasi di atas masih sulit karena banyaknya parameter: w , b , dan α_i . Untuk menyederhanakannya persamaan optimalisasi 14 harus ditransformasi ke dalam fungsi Langrange itu sendiri.

Persamaan Langrange Multiplier 13 dijabarkan menjadi

$$L_p = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i y_i (w \cdot x_i) - b \sum_{i=1}^N \alpha_i y_i + \sum_{i=1}^N \alpha_i$$
 (20)

Syarat optimal 16 ada dalam suku ke tiga ri ruas kanan dalam persamaan 20, dan memaksa suku ini menjadi sama dengan 0. Dengan mengganti w dari syarat 15, $\|w\|^2$ suku $w_i \cdot w_j$ persamaan di atas akan berubah menjadi dualitas Langrange Multiplier yang berupa L_d , di dapatkan:

Maksimalkan: $L_d = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$ (21)

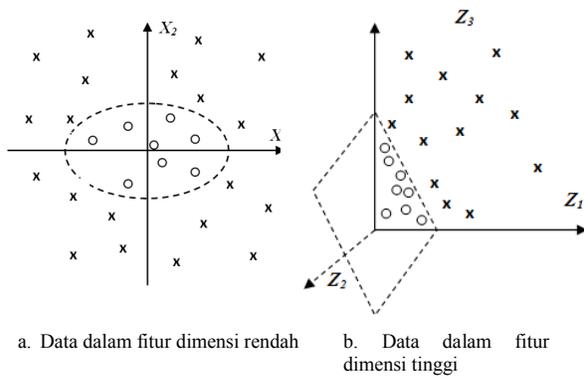
Syarat 1: $\sum_{i=1}^N \alpha_i y_i = 0$ (22)

Syarat 2: $\alpha_i \geq 0, i = 1, 2, \dots, N$ (23)

$X_i \cdot X_j$ merupakan *dot-product* dua data dalam data latih.

2.7. *SVM Non Linear*

Dalam SVM terdapat SVM linear dan SVM non linear (*kernel trick*). SVM sebenarnya hanya bekerja pada data yang dapat dipisahkan secara linear. Untuk data yang tidak linear dapat menggunakan metode kernel pada fitur data awal set data. Kernel adalah suatu fungsi yang memetakan fitur data dari dimensi rendah ke fitur baru dengan dimensi yang relatif lebih tinggi seperti diilustrasikan dalam gambar 6 (Prasetyo, 2014):



Gambar 6. Dimensi data (Prasetyo, 2014)

Algoritma pemetaan kernel dijelaskan sebagai berikut :

$$\Phi : D^r \rightarrow D^q$$

$$x \rightarrow \Phi(x)$$

Φ merupakan fungsi kernel yang digunakan untuk pemetaan, D merupakan data latih, r merupakan set fitur dalam satu data yang lama, q merupakan set fitur yang baru sebagai hasil pemetaan untuk setiap data latih. Sementara x merupakan data latih, dengan $x_1, x_2, \dots, x_N \in D^r$ merupakan fitur-fitur yang akan dipetakan ke fitur berdimensi tinggi q . Jadi set data yang digunakan sebagai pelatihan dengan algoritma adalah himpunan data dari dimensi fitur yang lama r ke dimensi bari q . Misalnya untuk contoh data N sebagai berikut:

$$(\Phi(x_1), y_1, \Phi(x_2), y_2, \dots, \Phi(x_N), y_N) \in D^q \quad (24)$$

Proses pemetaan pada fase ini memerlukan perhitungan perkalian titik dua buah data pada ruang fitur baru. perkalian titik kedua buah vektor (x_i) dan (x_j) dinotasikan sebagai $\Phi(x_i) \cdot \Phi(x_j)$, Nilai *dot-product* kedua buah vektor ini dapat dihitung secara tidak langsung, yaitu tanpa mengetahui fungsi transformasi Φ . Teknik komputasi seperti ini disebut *kernel trick*, yaitu menghitung *dot-product* dua buah vektor di ruang dimensi baru dengan memakai komponen kedua buah vektor tersebut di ruang dimensi asal sebagai berikut:

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (25)$$

Dan untuk prediksi pada set data dengan dimensi fitur yang baru di formulasikan sebagai berikut:

$$f(\Phi(z)) = \text{sign}(w \cdot \Phi(z) + b) = \text{sign} \left(\sum_{i=1}^N \alpha_i y_i \Phi(x_i) \cdot \Phi(z) + b \right) \quad (26)$$

N adalah jumlah data yang menjadi support vector, x_i adalah support vector, dan z adalah data uji yang akan dilakukan prediksi.

3. Metode Penelitian

Penelitian ini menggunakan GLCM untuk melakukan ekstraksi ciri, dan menggunakan metode *support vector machine* untuk pelatihan dan pengujian data. Data yang digunakan adalah citra daging kambing, daging kerbau, daging kuda, dan daging sapi yang diambil dengan jarak tinggi dari objek citra sekitar 20 cm, 30 cm, 40 cm.

3.1. Prosedur Penelitian

Prosedur penelitian klasifikasi citra jenis daging berdasarkan tekstur menggunakan ekstraksi ciri GLCM disajikan dalam Gambar 7. di bawah ini:



Gambar 7. Prosedur Penelitian

3.2. Algoritma Klasifikasi Citra

Algoritma klasifikasi citra jenis daging berdasarkan tekstur menggunakan ekstraksi ciri GLCM disajikan dalam Gambar 8.



Gambar 8. Algoritma Klasifikasi Citra

3.3. Algoritma Prapengolahan

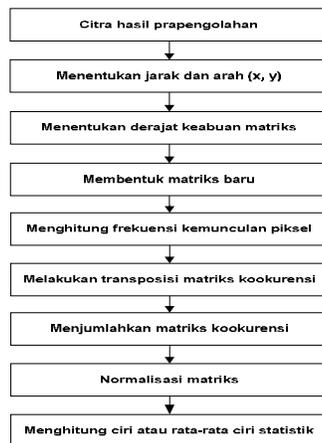
Prapengolahan dilakukan untuk memperoleh citra skala keabuan. Prapengolahan memiliki serangkaian proses yang ditunjukkan dalam Gambar 9.



Gambar 9. Algoritma Prapengolahan

3.4. Algoritma Ekstraksi Ciri Tekstur

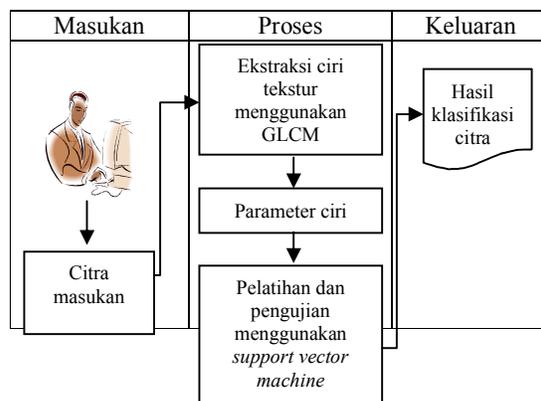
Algoritma ekstraksi ciri tekstur ditunjukkan dalam Gambar 10.



Gambar 10. Algoritma ekstraksi ciri tekstur menggunakan GLCM

3.5. Kerangka Sistem Informasi

Kerangka penelitian klasifikasi citra jenis daging berdasarkan tekstur menggunakan ekstraksi ciri GLCM disajikan dalam Gambar 11.



Gambar 11. Kerangka Penelitian

4. Hasil dan Pembahasan

Data yang digunakan pada penelitian ini adalah citra daging kambing, kerbau, kuda, dan sapi. Seluruh citra diambil menggunakan kamera digital Nikon D3200 dengan resolusi 24 mega piksel dengan jarak tinggi sekitar 20 cm, 30 cm, dan 40 cm dari objek citra. Jumlah masing-masing citra jenis daging tersebut adalah 150 citra. Dengan demikian, jumlah data citra daging untuk masing-masing jarak pengambilan adalah 600 citra. Sedangkan total keseluruhan citra adalah 1800 citra.

Dalam penelitian ini terdapat beberapa pembahasan yaitu pengaruh jarak piksel tetangga ($d=1, d=2, d=3, d=4$) dan arah GLCM terhadap hasil klasifikasi, pengaruh ukuran jarak pengambilan citra 20 cm, 30 cm, dan 40 cm terhadap hasil klasifikasi, pengaruh penggunaan nilai parameter yang berbeda menggunakan kernel RBF terhadap hasil klasifikasi.

4.1. Pengaruh jarak piksel tetangga dan arah GLCM terhadap hasil klasifikasi (citra dengan jarak pengambilan 20 cm)

Jarak piksel tetangga yang digunakan pada pembahasan ini adalah $d=1, d=2, d=3, d=4$. Sedangkan arah GLCM yang digunakan adalah empat arah yakni $0^\circ, 45^\circ, 90^\circ, 135^\circ$, dan gabungan keempat arah. Parameter sigma kernel RBF yang digunakan adalah 0,4. Citra yang digunakan adalah citra yang diambil dengan jarak 20 cm dengan jumlah 150 citra untuk masing-masing jenis daging. Total keseluruhan terdapat 600 citra. Jumlah citra yang digunakan sebagai pengujian untuk masing-masing jenis daging adalah 30 citra. Dengan demikian total keseluruhan data yang digunakan untuk pelatihan adalah 480 citra, dan total data untuk pengujian adalah 120 citra. Hasil pembahasan terhadap jarak piksel tetangga ini disajikan dalam Tabel 2.

Tabel 2. Tabel pengaruh jarak piksel tetangga dan arah GLCM dengan jarak pengambilan 20 cm

Jarak piksel tetangga	Jumlah data pengujian	Hasil Klasifikasi %									
		0°		45°		90°		135°		Gabungan 4 arah	
		B	S	B	S	B	S	B	S	B	S
$d=1$	120	75,8	24,2	73,3	26,7	78,3	21,7	84,2	15,8	78,3	21,7
$d=2$	120	69,2	30,8	73,3	26,7	75	25	87,5	12,5	79,2	20,8
$d=3$	120	70,8	29,2	73,3	26,7	75	25	75	25	77,5	22,5
$d=4$	120	70	30	70	30	78,3	21,7	82,5	17,5	76,7	23,3

Keterangan: B (benar), S (salah)

Berdasarkan Tabel 2 terlihat bahwa prosentase hasil klasifikasi dengan nilai *error* terkecil untuk keseluruhan jarak piksel tetangga berada pada jarak $d=2$ dengan arah GLCM 135° yaitu 87,5% untuk hasil klasifikasi benar dan 12,5% untuk hasil klasifikasi salah.

4.2. Pengaruh jarak piksel tetangga dan arah GLCM terhadap hasil klasifikasi (citra dengan jarak pengambilan 30 cm)

Pada pembahasan ini, citra yang digunakan adalah citra dengan jarak pengambilan 30 cm. Jarak piksel tetangga yang digunakan adalah $d=1, d=2, d=3, d=4$ dengan arah GLCM $0^\circ, 45^\circ, 90^\circ, 135^\circ$, dan gabungan keempat arah. Parameter sigma kernel RBF yang digunakan adalah 0,1. Total keseluruhan data adalah 600 citra (masing-masing jening daging 150 citra). Sedangkan total data yang digunakan untuk pengujian adalah 120 citra (masing-masing jenis daging 30 citra). Hasil pembahasan ini disajikan dalam Tabel 3.

Tabel 3. Tabel pengaruh jarak piksel tetangga dan arah GLCM dengan jarak pengambilan 30 cm

Jarak piksel tetangga	Jumlah data pengujian	Hasil Klasifikasi %									
		0°		45°		90°		135°		Gabungan 4 arah	
		B	S	B	S	B	S	B	S	B	S
d=1	120	75	25	72,5	27,5	81,7	18,3	73,3	26,7	73,3	26,7
d=2	120	79,2	20,8	70	30	83,3	16,7	70,8	29,2	71,7	28,3
d=3	120	75,8	24,2	66,7	33,3	80	20	75	25	75,8	24,2
d=4	120	74,2	25,8	61,7	38,3	78,3	21,7	65,8	34,2	75	25

Keterangan: B (benar), S (salah)

Dari Tabel 3 terlihat bahwa prosentase hasil klasifikasi dengan nilai *error* terkecil untuk keseluruhan jarak piksel tetangga berada pada jarak $d=2$ dengan arah GLCM 90° yaitu 83,3% untuk hasil klasifikasi benar dan 16,7% untuk hasil klasifikasi salah.

4.3. Pengaruh jarak piksel tetangga dan arah GLCM terhadap hasil klasifikasi (citra dengan jarak pengambilan 40 cm)

Citra yang digunakan adalah citra dengan jarak pengambilan 40 cm. Jarak piksel tetangga yang digunakan adalah $d=1, d=2, d=3, d=4$ dengan arah GLCM $0^\circ, 45^\circ, 90^\circ, 135^\circ$, dan gabungan keempat arah. Parameter sigma kernel RBF yang digunakan adalah 0,1. Total keseluruhan data adalah 600 citra. Sedangkan total data yang digunakan untuk pengujian adalah 120 citra. Hasil pembahasan ini disajikan dalam Tabel 4.

Tabel 4. Tabel pengaruh jarak piksel tetangga dan arah GLCM dengan jarak pengambilan 40 cm

Jarak piksel tetangga	Jumlah data pengujian	Hasil Klasifikasi %									
		0°		45°		90°		135°		Gabungan 4 arah	
		B	S	B	S	B	S	B	S	B	S
d=1	120	66,7	33,3	66,7	33,3	71,7	28,3	69,2	30,8	69,2	30,8
d=2	120	70,8	29,2	69,2	30,8	80	20	66,7	33,3	69,2	30,8
d=3	120	65,8	34,2	70	30	77,5	22,5	69,2	30,8	75	25
d=4	120	60	40	60,8	39,2	70	30	67,5	32,5	73,3	26,7

Keterangan: B (benar), S (salah)

Dari Tabel 4 terlihat bahwa prosentase hasil klasifikasi dengan nilai *error* terkecil untuk keseluruhan jarak piksel tetangga berada pada jarak $d=2$ dengan arah GLCM 90° yaitu 80% untuk hasil klasifikasi benar dan 20% untuk hasil klasifikasi salah.

4.4. Pengaruh ukuran pengambilan citra 20 cm, 30 cm, dan 40 cm terhadap hasil klasifikasi

Pembahasan ini akan membandingkan citra yang diambil dengan jarak 20 cm, 30 cm, dan 40 cm. Jarak piksel tetangga yang digunakan adalah $d=1$ dengan arah GLCM $0^\circ, 45^\circ, 90^\circ, 135^\circ$, dan gabungan keempat arah. Parameter sigma kernel RBF yang digunakan adalah 0,3. Total keseluruhan terdapat 600 citra. Jumlah citra yang digunakan sebagai pengujian adalah 120 citra. Hasil pembahasan ini disajikan dalam Tabel 5.

Tabel 5. Tabel ukuran jarak pengambilan citra 20 cm, 30 cm, dan 40 cm terhadap hasil klasifikasi.

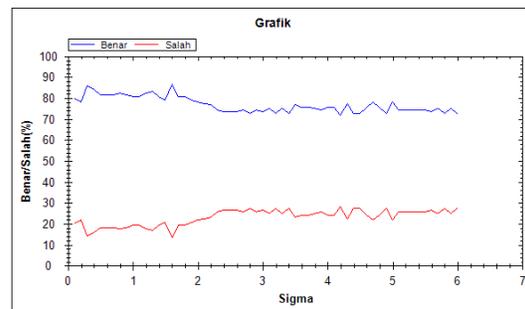
Jarak pengambilan citra	Jumlah data pengujian	Hasil Klasifikasi %									
		Jarak piksel tetangga d=1									
		0°		45°		90°		135°		Gabungan 4 arah	
		B	S	B	S	B	S	B	S	B	S
20 cm	120	74,2	25,8	70	30	78,3	21,7	85,8	14,2	76,7	23,3
30 cm	120	70	30	72,5	27,5	75	25	69,2	30,8	71,7	28,3
40 cm	120	63,3	36,7	66,7	33,3	65	35	66,7	33,3	60	40

Keterangan: B (benar), S (salah)

Pada Tabel 5 terlihat bahwa prosentase dengan nilai *error* terkecil terletak pada jarak pengambilan 20 cm pada arah GLCM 135° yaitu sebesar 85,8% untuk hasil klasifikasi benar dan 14,2% untuk hasil klasifikasi salah.

4.6. Pengaruh penggunaan nilai parameter yang berbeda terhadap hasil klasifikasi

Pada Tabel 5 jika jarak pengambilan citra 20 cm dengan jarak piksel tetangga $d=1$ pada arah GLCM 135° menggunakan parameter sigma kernel RBF 0,1 sampai dengan 6,0, maka hasil pengujian dapat dilihat pada Gambar 12.

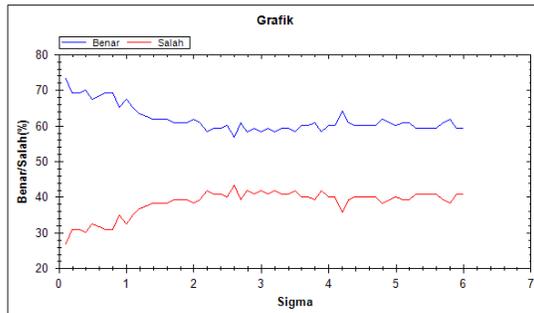


Gambar 12. Grafik hasil klasifikasi dengan jarak pengambilan 20 cm

Dari Gambar 12 terlihat bahwa dengan menggunakan parameter sigma yang berbeda-beda,

nilai *error* terkecil untuk hasil klasifikasi adalah 14,2% yakni pada parameter sigma 0,3.

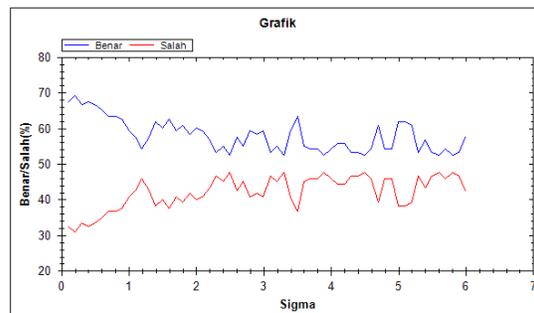
Jika jarak pengambilan citra 30 cm dengan jarak piksel tetangga $d=1$ pada arah GLCM 135^0 pada tabel 4.4 menggunakan parameter sigma kernel RBF 0,1 sampai dengan 6,0, maka hasil pengujian dapat dilihat pada Gambar 13.



Gambar 13 Grafik hasil klasifikasi dengan jarak pengambilan 30 cm

Dari Gambar 13 terlihat bahwa dengan menggunakan parameter sigma 0,1 sampai dengan 6,0, nilai *error* terkecil untuk hasil klasifikasi adalah 26,7% yakni pada parameter sigma 0,1.

Jika jarak pengambilan citra 40 cm dengan jarak piksel tetangga $d=1$ pada arah GLCM 135^0 pada tabel 4.4 menggunakan parameter sigma kernel RBF 0,1 sampai dengan 6,0, maka hasil pengujian dapat dilihat pada Gambar 14.



Gambar 14. Grafik hasil klasifikasi dengan jarak pengambilan 40 cm

Dari Gambar 14 terlihat bahwa dengan menggunakan parameter sigma yang berbeda-beda, nilai *error* terkecil untuk hasil klasifikasi adalah 30,8% yakni pada parameter sigma 0,2.

5. Kesimpulan

Dari hasil penelitian ini, dapat diambil kesimpulan bahwa:

1. Sistem pengenalan daging yang dikembangkan, memiliki tingkat pengenalan terbaik 87,5% yang terjadi pada citra dengan jarak pengambilan 20 cm dengan kondisi jarak piksel tetangga $d=2$

pada arah GLCM 135^0 . Tingkat pengenalan tersebut berada pada parameter sigma 0,4.

2. Citra dengan ukuran jarak pengambilan 20 cm memiliki nilai *error* paling kecil yakni 14,2% dibandingkan dengan citra yang diambil dengan jarak 30 cm ataupun 40 cm. Nilai *error* tersebut berada pada jarak piksel tetangga $d=1$ dengan arah GLCM 135^0 .
3. Penggunaan nilai parameter sigma kernel RBF 0,1 sampai dengan 6,0 untuk masing-masing citra dengan jarak pengambilan 20 cm, 30 cm, dan 40 cm, menghasilkan klasifikasi dengan nilai *error* terkecil yakni 14,2%. *Error* terkecil tersebut berada pada citra dengan jarak pengambilan 20 cm dengan jarak piksel tetangga $d=1$ pada arah GLCM 135^0 .
4. Berdasarkan berbagai pengujian yang telah dilakukan, citra dengan ukuran jarak pengambilan 20 cm memiliki tingkat pengenalan terbaik dibandingkan dengan citra yang diambil dengan jarak 30 cm ataupun 40 cm. Dengan demikian, citra dengan ukuran jarak pengambilan 20 cm dapat direkomendasikan untuk klasifikasi citra jenis daging.

Daftar Pustaka

- Basset, O., Buquet, B., Abouelkaram, S., Delachartre, P., dan Culioli, J., 2000. Application of texture image analysis for the classification of bovine meat. *Food Chemistry*, 69, 437-445.
- Bharati, M. H., Liu, J. J. and Mac Gregor, J. F., 2004. Image texture analysis: methods and comparisons. *Chemometrics and Intelligence Laboratory Systems*, 72, 57-71.
- Burges, C.J., 1998. A Tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2, 1-43.
- Haralick, R.M., 1979. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67, 786-804.
- Haralick, R.M., Shanmugam, K., dan Dinstein, I., 1973. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 6, 610-621.
- Kadir, A., dan Susanto, A., 2012. Teori dan aplikasi pengolahan citra, Andi: Yogyakarta.
- Kadir, A. dan Susanto, A., 2013. Teori dan Aplikasi Pengolahan Citra, Andi: Yogyakarta.
- Kudo, M. and Sklansky, J., 2000. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33, 25-41.
- Krutz, G.W., Gibson, H.G., Cassens, D.L. and Zhang, M., 2000. Colour vision in forest and wood engineering. *Landwards*, 55, 2-9.
- Nello, C., dan Taylor, J.S., 2000. *An introduction to support vector machines and other kernel-*

- based learning methods*, Cambridge University Press.
- Patel, D., Davies, E.R. and Hannah, I, 1996. The use of convolution operators for detecting contaminants in food images. *Pattern Recognition*, 29 (6), 1019–1030.
- Prasetyo, E., 2012. *Data Mining Konsep dan Aplikasi Menggunakan Matlab*, Andi: Yogyakarta.
- Prasetyo, E., 2014. *Data Mining Mengolah Data Menjadi Informasi Menggunakan Matlab*. Andi Publisher: Yogyakarta.
- Salat, R. dan Osowski, S., 2004. Accurate fault location in the power transmission line using support vector machine approach, *IEEE Trans. Power Syst.*, 19, 879–886.
- Seetha, M., Muralikrishna, I.V., Deekshatulu, B.L., Malleswari, B.L., Nagaratna, Hedge, P., 2008. Artificial neural network and other methods of image classification. *Journal of Theoretical and Applied Information Technology*, 1039–1053.
- Schwartz. W, Pedrini. H, 2006. Textured image segmentation based on spatial dependence using a Markov random field model, *IEEE International Conference on Image Processing*, Atlanta, GA, USA, pp. 2449–2452.
- Sun, D-W., 2000. Inspecting pizza topping percentage and distribution by a computer vision method. *Journal of Food Engineering*, 44, 245–249.
- Siqueira, F.R.D., Schwartz, W.R., Pedrini, H., 2013. Multi-scale gray level co-occurrence matrices for texture description. *Neurocomputing*, 120, 336–345.
- Zheng, C., Sun, D-W., dan Zheng, L., 2006. Recent applications of image texture for evaluation of food qualities—a review. *Trends in Food Science & Technology*, 17, 113–128.