

KLASIFIKASI BOBOT KARKAS DOMBA BERDASARKAN METODE FRACTAL DAN KLASIFIKASI K-NEAREST NEIGHBOUR

CLASSIFICATION OF SHEEP CARCASS WEIGHT BASED ON FRACTAL METHODS AND CLASSIFICATION K-NEAREST NEIGHBORS

Annisa Adlina Mulyaningrum¹, Dr. Ir. Bambang Hidayat, DEA², Nur Ibrahim, S.T,M.T³

^{1,2}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
¹annisaadlina@student.telkomuniversity.ac.id, ²bhidayat@telkomuniversity.ac.id,
³nuribrahim.nib@gmail.com

Abstrak

Domba adalah hewan ternak yang banyak dimanfaatkan oleh masyarakat di Indonesia. Salah satu pemanfaatan dari domba adalah karkas domba, dimana permintaannya terus meningkat dari waktu ke waktu. Karkas domba adalah berat bersih daging tanpa kepala, kaki, organ dalam dan juga ekor. Saat ini penimbangan terhadap domba masih dilakukan secara manual, yaitu dengan cara menggendong hewan tersebut, menduga dengan cara melihatnya dan menduga secara subjektif. Karena rentan bobot domba tersebut memiliki berat di bawah 40 kg. Namun cara tersebut hanya berlaku untuk pembeli yang sudah berpengalaman, namun berpotensi merugikan pembeli amatir atau baru.

Image processing atau pengolahan citra digital diimplementasikan untuk membantu mengestimasi bobot karkas domba. Untuk mendapatkannya pengolahan citra dilakukan dengan cara memisahkan tubuh domba dengan *background*. Setelah itu, proses identifikasi untuk mendapatkan ukuran fisik (lingkar dada dan panjang badan). Tugas akhir ini bertujuan untuk merancang sistem berbasis Matlab, berfungsi mempermudah klasifikasi karkas domba menjadi 3 kelas bagian, membuktikan rumus perhitungan Ardjodarmoko dapat digunakan untuk estimasi karkas domba. Pertama sistem akan melakukan ekstraksi ciri untuk mendapatkan informasi di dalam citra menggunakan metode *Fractal*. Setelah didapatkan ciri citra, kemudian akan diklasifikasikan dengan metode jarak terdekat atau *K-Nearest Neighbor* yang kemudian akan dikelompokkan sesuai kelasnya besar, sedang, dan kecil.

Dari hasil pengujian diperoleh nilai akurasi sistem sebesar 83,33% dengan waktu komputasi 0,49 detik, menggunakan metode ekstraksi ciri *Fractal* dengan jumlah matriks ciri sama dengan 6, parameter $s = [2\ 4\ 8\ 16\ 32\ 64]$. Sedangkan proses klasifikasi *K-Nearest Neighbor* aturan jarak paling baik yang digunakan adalah *euclidean* yang menghasilkan nilai akurasi terbaik 83,33% dengan parameter terbaik pada nilai $K=1$, dan sistem memiliki tingkat *error* sebesar 16, 67%.

Kata Kunci : Karkas Domba, *Fractal*, *K-Nearest Neighbor*(KNN)

Abstract

Sheep are farm animals that are widely used by people in Indonesia. One of the uses of sheep is sheep carcass, where demand continues to increase over time. Sheep carcass is a net weight of meat without head, legs, internal organs and tail. Currently weighing the sheep is still done manually, that is by carrying the animal, suspect by looking at it and guessed subjectively. Because the vulnerable weight of the sheep has a weight below 40 kg. But this method only applies to experienced buyers, but has the potential to harm amateur or new buyers.

Image processing or digital image processing is implemented to help estimate carcass weight of sheep carcass. To get the image processing done by separating the body of the sheep from the background. After that, the identification process to get the physical size (chest circumference and body length). This final project aims to design Matlab-based system, to facilitate the classification of sheep carcass into 3 parts class, proves the calculation formula of Ardjodarmoko can be used to estimate sheep carcass. First the system will extraction the feature to get information in the image using Fractal method. Once the image is obtained, it will then be classified by the closest distance method or K-Nearest Neighbor which will then be grouped according to its large, medium, and small class.

From the test results obtained system accuracy value of 83.33% with a computational time of 0.49 seconds, using Fractal characteristic extraction method with the number of characteristic matrix equal to 6, $s = [2\ 4\ 8\ 16\ 32\ 64]$. While the K-Nearest Neighbor classification process, the best distance rule used is euclidean which yields the best accuracy value of 83.33% with the best parameter at the value of $K = 1$, and the system has an error rate of 16, 67%.

Keywords : *sheep, Carcass, Fractal, K-Nearest Neighbor*

1. Pendahuluan

Domba merupakan hewan ternak yang banyak diminati. Selain itu domba menjadi hewan favorit untuk di pelihara karena mudahnya dalam memelihara domba. Bobot badan domba merupakan salah satu indikator produktivitas ternak yang dapat diduga berdasarkan ukuran *linear* tubuh domba meliputi lingkar dada, panjang badan dan tinggi badan. Peternak umumnya menggunakan bobot hidup domba sebagai ukuran keberhasilan pemeliharaan dan pertumbuhannya. Bobot hidup juga menjadi pertimbangan penentu harga seekor domba potong dalam bidang pemasaran.

Perkembangan era yang semakin maju namun tetap saja sebagian besar pedagang hanya melihat visual dari domba, beberapa menggunakan alat timbang konvensional yang dimana alat tersebut harus dibawa kemana-saja. Selain itu, teknik pengukuran berat karkas domba lainnya yaitu dengan menggendongnya, Namun hal tersebut tidak lah efisien, karena kemungkinan kekeliruan masih bisa terjadi dan cukup menyusahakan untuk peternak yang menginginkan kepastian harga. Seiring pesatnya perkembangan teknologi di Indonesia, berdasarkan pemaparan latar belakang diatas, maka dibuat sistem untuk mengidentifikasi berat karkas bobot domba dengan menggunakan *image processing*.

Dalam penelitian ini penulis menggunakan metode *Fractal* untuk mendapatkan karakteristik ciri dari citra digital, karena fraktal merupakan salah satu algoritma yang digunakan untuk mengklasifikasi objek yang memiliki bentuk ketidakteraturan dan kemudian akan di klasifikasikan menggunakan metode *K-Nearest Neighbor*.

Dari Penelitian ini telah dihasilkan sebuah alat bantu klasifikasi bobot karkas domba yang memiliki keakuratan tinggi dan waktu komputasi yang singkat.

2. Dasar Teori

2.1 Domba

Karkas merupakan bagian tubuh dari seekor domba yang sering dimanfaatkan oleh manusia, meliputi bagian tubuh domba yang telah dikuliti, tanpa jeroan, kepala, keempat kaki (mulai dari *karpus* dan *tarsus*), organ reproduksi, ambing, ekor dan lemak yang berlebih. [1]



Gambar 1. Domba

Bobot karkas domba dapat dihitung dengan mengalikan bobot hidup domba dengan presentase karkas yang telah ditentukan. Presentase bobot karkas domba terhadap bobot hidupnya berkisar antara 40-44%. Bobot domba hidup dapat dihitung dengan menggunakan Rumus Ardjodarmoko yang telah ditetapkan oleh para ahli. Berikut adalah Rumus Ardjodarmoko[2] :

$$BB = \frac{(LD^2) \times (PB)}{10^4} \quad (1)$$

Keterangan :

BB = Bobot Badan Sapi Hidup (kg)

LD = Lingkar Dada Sapi (cm)

Rumus ini merupakan penyempurnaan dari rumus *winter*, yang diaplikasikan pada domba. Kalkulasi bobot karkas yang penulis gunakan yaitu 43.20% dari bobot domba hidup. Sehingga rumus estimasi bobot karkas domba direpresentasikan pada persamaan (2) :

$$BK = 0.4320 \times BB \quad (2)$$

Keterangan:

BK = Berat Karkas (kg)

BB = Berat Badan (kg)

2.2 Citra Digital

Pengolahan citra digital adalah sebuah ilmu yang mempelajari tentang teknik-teknik pengolah citra. Citra merupakan fungsi kontinyu dengan intensitas cahaya pada bidang dua dimensi. Reperesentasi dari fungsi kontinyu menjadi nilai-nilai diskrit disebut digitalisasi citra. Sebuah citra digital merupakan fungsi dari dua *variable* yaitu $f(x,y)$ dimana koordinat x menyatakan baris dan koordinat y menyatakan kolom sedangkan $f(x,y)$ merupakan derajat keabuan pada (x,y) dimana berukuran dari M kolom dan N baris, masing- masing elemen pada sebuah citra digital disebut dengan piksel. Citra digital dalam bentuk matrik dapat di presentasikan sebagai berikut:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M-1) \\ f(1,0) & f(1,1) & \dots & f(1, M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1, M-1) \end{bmatrix} \quad (3)$$

Keterangan:

N = jumlah baris, $0 <= y <= N-1$

M = jumlah kolom, $0 <= x <= M-1$

Besarnya nilai M , N dan G pada umumnya merupakan perpangkatan dari dua

$$M = 2^m; N = 2^n; G = 2^k \quad (4)$$

Nilai m , n dan k adalah bilangan bulat positif. Pembagian skala $(0, G)$ menjadi G level yang dinyatakan dengan suatu harga bilangan *integer*, besar G tergantung pada proses digitalisasinya. Biasanya keabuan 0 menyatakan intensitas hitam dan 1 menyatakan intensitas putih. Untuk citra 8 bit, nilai G sama dengan $2^8 = 256$ warna (derajat keabuan) [3].

2.2 Dimensi Fraktal

Dimensi fraktal merupakan karakteristik penting karena berisikan informasi struktur geometri. Apabila diperbesar akan terlihat benda geometri serupa yang menyusun benda itu sendiri atau yang biasa disebut *self-similarity*. Dimensi fraktal merupakan pengukuran efektif untuk objek yang kompleks, dimana dapat di aplikasikan pada segmentasi citra dan pengenalan bentuk.[4] Metode umum yang biasa di gunakan untuk menghitung dimensi dari objek fraktal adalah metode perhitungan kotak (box counting) dapat di persamaan berikut :[5]

$$D(s) = \frac{\log(N(s))}{\log s} \quad (5)$$

Keterangan :

$N(s)$ = jumlah kotak yang berukuran s , berisi nilai ciri dari citra (piksel objek)

$D(s)$ = dimensi fraktal dengan kotak yang berukuran s .

Jika terdapat beberapa macam ukuran kotak yang direpresentasikan, maka metode perhitungan kotak dilakukan dengan langkah-langkah berikut :

1. Membagi citra ke dalam kotak-kotak berukuran s .
2. Menghitung banyaknya kotak ($N(s)$) yang dibutuhkan untuk mencakup suatu objek. Nilai $N(s)$ bergantung pada s . Variabel s memiliki nilai yang berubah dari 1 hingga 2^k ($k = 0, 1, 2, \dots$) dan 2^k tidak boleh bernilai lebih besar dari ukuran citra. Maka jika citra berukuran $2^m \times 2^n$, nilai k akan berhenti hingga m .

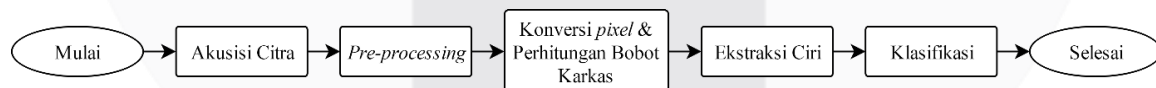
2.3 K-Nearest Neighbor [6][7]

K-Nearest Neighbour (KNN) adalah mengklasifikasikan citra uji ke dalam kelas dengan jumlah anggota terbanyak. Prinsip kerja KNN adalah mencari jarak terdekat antara data yang akan dievaluasi dengan k tetangga (neighbor) terdekatnya dalam data pelatihan. Data pelatihan diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data pelatihan. Sebuah titik pada ruang ini ditandai kelas c , kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan jarak. Berikut adalah rumus perhitungan jarak dengan aturan *Euclidean Distance* :

$$L_2(X, Y) = \sqrt{\sum_{i=1}^d (X_i - Y_i)^2} \quad (6)$$

3. Desain Sistem

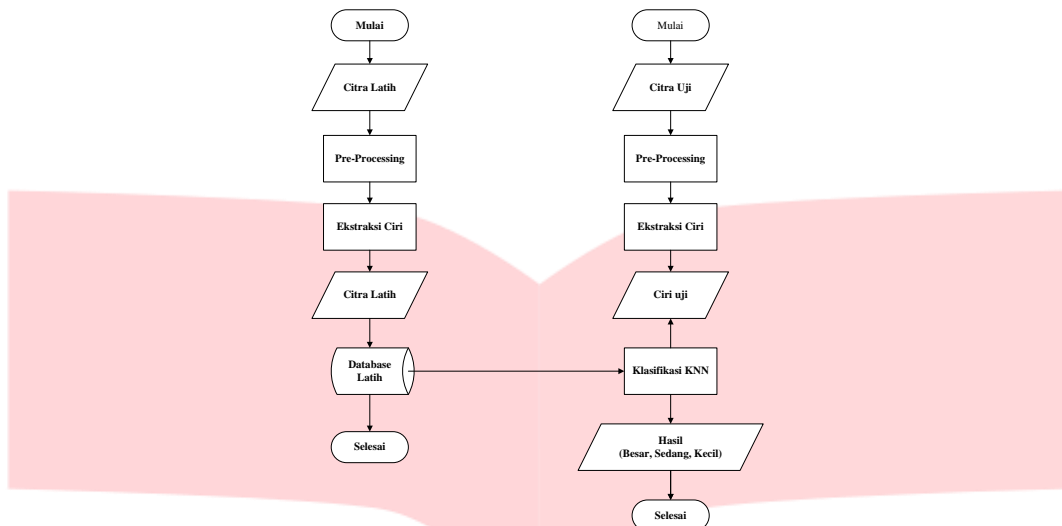
Gambaran umum dari sistem yang dirancang pada penelitian kali ini direpresentasikan oleh diagram alir berikut:



Gambar 2. Diagram Alir Sistem

3.1 Diagram Alur Rancangan Sistem

Dalam merancang sistem pada proram aplikasi estimasi bobot karkas domba berbasis pengolahan citra digital menggunakan Matlab, dilakukan beberapa tahapan yaitu: tahap pelatihan dan tahap pengujian. Citra latih akan diawali dengan tahapan akuisisi citra domba, kemudian mengalami tahapan perbaikan kualitas citra (*pre-processing*), tahapan selanjutnya konversi pixel perhitungan & bobot karkas, dan mengalami proses pembuatan *database*. Di lanjutkan pada tahap uji tersebut akan di tahapan ekstraksi ciri, dan tahapan klasifikasi.



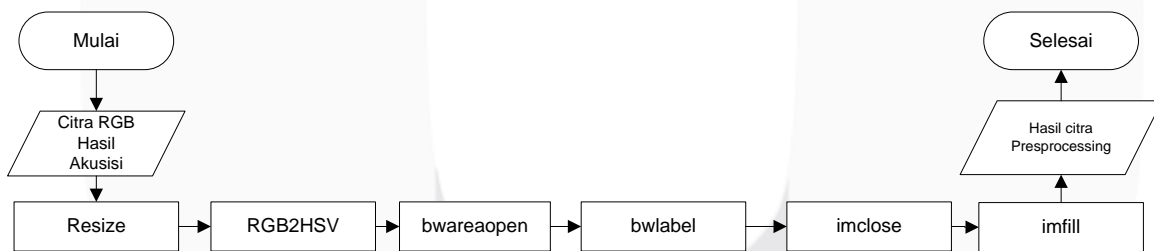
Gambar 3. Diagram Alir Proses Latih (kiri) dan Proses Uji (Kanan)

3.2 Akuisisi Citra

Akuisisi citra merupakan proses dasar atau awal dalam mendapatkan citra digital. Domba yang di gunakan merupakan domba milik Rumah Pemotongan Hewan (RPH) Aqiqah. Kamera yang di gunakan pada proses ini adalah kamera DSLR dengan jarak 1,5 meter. Citra domba di ambil dari tampak sisi samping. Format gambar domba yang di peroleh dari pengambilan gambar adalah (*.JPG).

3.3 Pre-processing

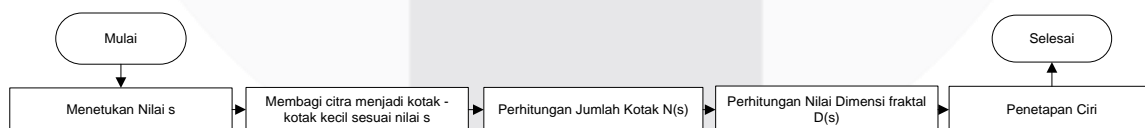
Pre-processing adalah tahap selanjutnya untuk memperbaiki kualitas citra yang telah di ambil. Tujuan dari *pre-processing* untuk meningkatkan kualitas citra yang lebih bagus dari citra sebelumnya sehingga dapat mempermudah pengolahan citra selanjutnya seperti ekstraksi ciri dan klasifikasi. Tahapan *pre-processing* dapat dilihat pada Gambar 4 di awali dengan *resize* citra dengan *ratio* yang di tentukan untuk mengubah ukuran citra sehingga mendapatkan hasil yang segmentasi yang baik dan mengurangi waktu komputasi program. Kemudian citra RGB yang sudah di *resize* di transformasikan ke dalam ruang warna HSV. Kemudian citra melalui proses *threshol* untuk segmentasi, memisahkan objek dengan background. Kemudian mengalami proses filter morfologi (*bwareopen*) dengan tujuan melakukan proses pembukaan secara morfologikal. Setelah itu mengalami proses *bwdist* untuk menghilangkan objek-objek kecil pada citra biner, *labelling* untuk memberi label pada bagian terbesar dari citra biner, proses morfologi closing (*imclose*) untuk melakukan operasi penutupan secara morfologikal, dan proses morfologi filling (*imfill*) untuk mengisi lubang sertas region dalam citra.



Gambar 4. Diagram Alir *Pre-processing*

3.4 Ekstraksi Ciri

Ekstraksi Ciri merupakan suatu cara atau teknik untuk mengambil ciri pada citraguna melalukan pengenalan pola. Pada tahapan ini setelah proses *Pre-processing* selesai. Di lakukan proses ekstraksi ciri dengan metode fraktal untuk mendapatkan parameter ciri pemilik citra yang kemudian akan di gunakan untuk menghitung lingkard dada domba dan panjang badan domba. Berikut adalah diagram alir proses ekstraksi ciri:



Gambar 5. Diagram Alir Proses Ekstraksi ciri

3.5 Klasifikasi

Proses klasifikasi adalah proses pengambilan ciri terhadap citra, dimana proses yang akan dilakukan dengan cara memisahkan citra menjadi citra latih dan citra uji. Proses klasifikasi merupakan proses lanjut dari hasil proses ekstrasi ciri. Semua informasi yang sudah tercatat akan diklasifikasikan dengan menggunakan *K-Nearest Neighbor* dengan menggunakan ketentuan jarak seperti *euclidean distance*, *cityblock distance*, *cosine*

distance, dan *correlation distance*. Pada dasarnya algoritma ini akan menghitung jarak antara nilai ciri yang dimiliki oleh data uji menuju data latih. Lalu, jarak-jarak tersebut akan dipilih jarak terpendeknya. Setelah itu, klasifikasi mengeluarkan hasil domba termasuk berukuran apa.

3.6 Performansi Sistem

Pada tahap Performansi sistem ini bertujuan untuk mengetahui kinerja dari sebuah sistem apakah berjalan dengan baik atau tidak. Berikut persamaan yang di gunakan untuk perhitungan:

1. Akurasi Sistem

Merupakan ukuran ketepatan suatu sistem dalam mengenali yang di berikan sehingga dapat menghasilkan keluaran yang benar. Direpresentasikan menggunakan persamaan sebagai berikut:

$$\text{Akurasi} = \frac{\text{Jumlah data benar}}{\text{Jumlah data keseluruhan}} \times 100\% \quad (7)$$

2. Waktu Komputasi

Merupakan waktu yang diperlukan oleh suatu sistem untuk melakukan sebuah proses. Pada sistem, waktu komputasi yang dihitung adalah waktu selesai dikurangkan waktu mulai. Secara matematis direpresentasikan menggunakan persamaan berikut:

$$\text{Waktu Komputasi} = \text{Waktu selesai} - \text{Waktu mulai} \quad (8)$$

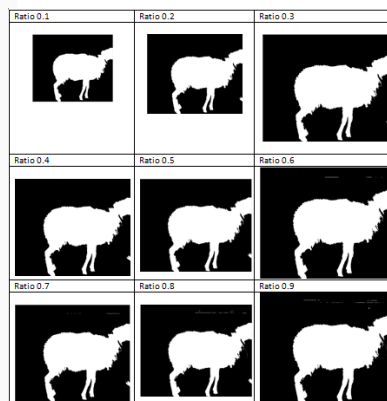
4. Hasil Pengujian Sistem

4.1 Hasil Pengujian Perubahan Nilai Ratio pada Proses Resize

Untuk mengetahui pengaruh dari ukuran citra terhadap akurasi dari sistem maka akan ada beberapa skenario yang diujikan ke sistem. Pengujian ini dilakukan menggunakan 78 citra uji dan 234 citra latih, sistem akan diuji dengan perubahan nilai *ratio* citra pada proses *resize*. Perubahan nilai *ratio* dimulai dari 0.1 hingga 0.9. Nilai *s* matriks ciri adalah $s = [2 \ 4 \ 8]$, nilai parameter *K-Nearest Neighbor* yang digunakan adalah $k = 1$ untuk mempercepat waktu komputasi dan *distance* yang digunakan adalah *euclidean*. selanjutnya akan dilakukan analisis pengaruh perubahan nilai *ratio* terhadap hasil nilai pada *pre-processing*. Berikut adalah hasil dari perubahan nilai *ratio* :



Gambar 6. Citra Masukan



Gambar 7. Hasil Segmentasi *Pre-processing* dengan Perubahan Nilai *Ratio*

Dari Gambar 7 dapat diketahui bahwa tiap nilai *ratio* pada *resize* yang digunakan akan mempengaruhi hasil *pre-processing*. Nilai *ratio* akan mempengaruhi ukuran citra masukan, semakin kecil nilai *ratio* maka ukuran citra masukan akan semakin kecil. Gambar 7 menunjukkan bahwa nilai *ratio* yang kecil dapat menyebabkan hasil menjadi buruk karena citra memiliki resolusi terlalu kecil. dan terlihat juga bahwa kualitas hasil meningkat seiring bertambahnya nilai *ratio* rentan 0.1 hingga 0.3 akan tetapi pada rentang nilai *ratio* 0.4 hingga 0.9 hasil terlihat kurang baik, nilai *ratio* apabila terlalu besar juga tidak baik karena *resize* bekerja dengan cara *down sampling* yaitu mengurangi jumlah *pixel* dan menghilangkan sebagian informasi dari citra. Karena menghilangkan sebagian informasi dari citra, informasi yang dihasilkan oleh *down sampling* akan menjadi kurang optimal. Maka dapat disimpulkan bahwa hasil hasil yang paling baik dan sesuai dengan tujuan *pre-processing* ditunjukkan pada nilai *ratio* sebesar 0.3.

Tabel 1 Perubahan Nilai Ratio Terhadap Akurasi dan Waktu Komputasi

No	Ratio	Akurasi (%)	Waktu komputasi (s)
1	0.1	76.92	0.23
2	0.2	73.08	0.53
3	0.3	80.76	0.45
4	0.4	78.21	0.44
5	0.5	79.49	0.94
6	0.6	76.92	0.44
7	0.7	76.33	0.43
8	0.8	75.64	0.39
9	0.9	74.36	0.33

Berdasarkan Tabel 1 dapat diketahui bahwa perubahan nilai *ratio* pada *resize* akan mempengaruhi waktu komputasi sistem. Pada nilai *ratio* 0.1 sampai dengan 0.5 waktu komputasi bertambah atau naik, namun pada nilai *ratio* 0.6 sampai dengan 0.9 waktu komputasi berkurang atau menurun. Pada skenario ini di dapatkan akurasi paling tinggi pada *ratio* 0.3 dengan akurasi tertinggi 80.76% dan dengan waktu komputasi 0.44 detik.

4.2 Hasil Pengujian Skenario 2

Pada skenario 2 dilakukan untuk menganalisis pengaruh perubahan ukuran matriks ciri dimensi fraktal terhadap akurasi dan waktu komputasi sistem. Nilai dimensi fraktal didapatkan dari perhitungan persamaan (5) yang dipengaruhi oleh parameter s (2^k). Ukuran matriks ciri tergantung pada jumlah dimensi fraktal yang dihitung dari setiap citra. Nilai *ratio* 0.3 sesuai dengan hasil dari skenario 1, nilai parameter *K-Nearest Neighbor* yang digunakan adalah $k = 1$ untuk mempercepat waktu komputasi dan *distance* yang digunakan adalah *euclidean*. Hasil pengujian skenario 2 ditunjukkan pada Tabel 2.

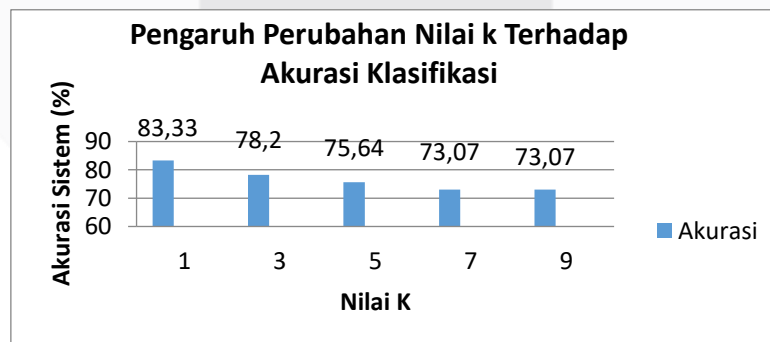
Tabel 2 Pengaruh Perubahan Jumlah Dimensi Fraktal Terhadap Akurasi Sistem

Nilai s Matriks Ciri	Jumlah Dimensi Fraktal	Akurasi Sistem (%)	Waktu Komputasi (detik)
[2 4 8]	3	82,05	0,24
[2 4 8 16]	4	82,05	0,36
[2 4 8 16 32]	5	82,05	0,54
[2 4 8 16 32 64]	6	83,33	0,67

Berdasarkan Tabel 2 dapat dilihat bahwa perubahan ukuran matriks ciri dan jumlah dimensi fraktal mempengaruhi klasifikasi sistem dalam proses pengklasifikasi domba. ukuran matriks ciri atau jumlah dimensi fraktal mempengaruhi waktu komputasi sistem. Berdasarkan perubahan nilai jumlah dimensi fraktal dari 3 hingga 6 dapat disimpulkan bahwa semakin banyak atau besar jumlah dimensi fraktal yang dihitung pada proses ekstraksi ciri akan meningkatkan waktu komputasi sistem. Hasil akurasi paling baik di dapat pada jumlah dimensi fraktal 6 dengan akurasi sebesar 83.33%.

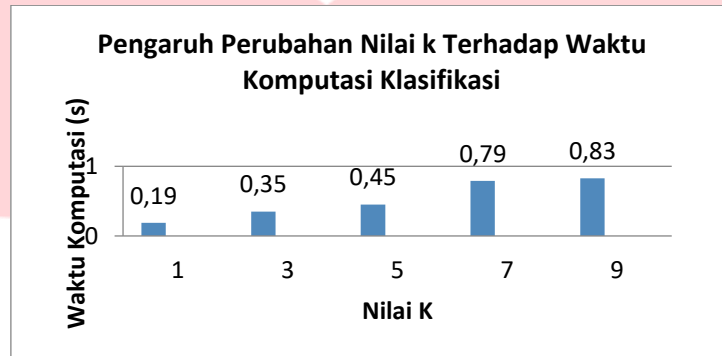
4.3 Hasil Pengujian Skenario 3

Pada pengujian skenario 3, dilakukan analisis pengaruh k dalam metode *K-Nearest Neighbor* terhadap akurasi dan waktu komputasi sistem. Akurasi yang dihitung pada pengujian kali ini adalah akurasi sistem dalam proses klasifikasi domba berdasarkan *database*. pengujian ini menggunakan jarak atau *distance K- Nearest Neighbor Euclidean*, *ratio* 0.3 dan ukuran matriks ciri dimensi fraktal $s = [2\ 4\ 8\ 16\ 32\ 64]$. Perubahan k yang di gunakan yaitu 1, 3, 5, 7 dan 9. Penguji skenario dilakukan pada 234 citra latih dan 78 citra uji



Gambar 8. Grafik Pengaruh Nilai k Terhadap Akurasi Klasifikasi

Berdasarkan Gambar 8 dapat dilihat bahwa perubahan nilai k mempengaruhi akurasi sistem dalam proses klasifikasi, hal ini dikarenakan semakin besar nilai k maka semakin banyak tetangga yang digunakan untuk proses klasifikasi dan kemungkinan untuk terjadinya noise juga semakin besar ditambah lagi dengan adanya dominasi atau frekuensi kelas data latih yang tidak seimbang dari suatu kelas tertentu sehingga hasilnya data cenderung diklasifikasikan pada data kelas yang mendominasi. Pada pengujian yang dilakukan nilai akurasi maksimum cenderung terjadi saat nilai $k = 1$ atau $k \leq 5$. Semakin kecil nilai k berarti semakin sedikit jumlah tetangga yang digunakan untuk proses klasifikasi data baru. Akurasi paling tinggi pada $k=1$ yaitu dengan nilai 83,33%.Selanjutnya hasil pengujian pengaruh perubahan nilai k terhadap waktu komputasi direpresentasikan oleh Gambar 9 berikut ini :

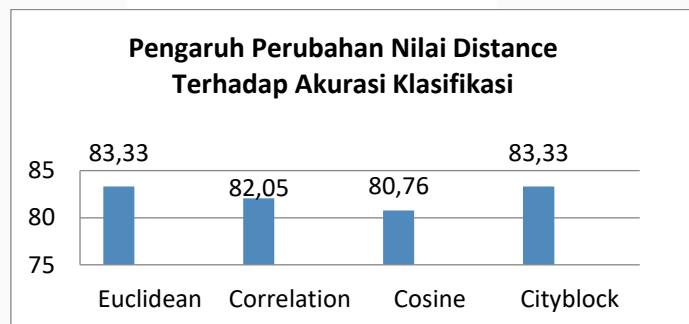


Gambar 9. Grafik Pengaruh Nilai k Terhadap Waktu Komputasi Klasifikasi

Berdasarkan Gambar 9 dapat dilihat bahwa perubahan nilai k mempengaruhi waktu komputasi sistem. Semakin besar nilai k pada metode K -Nearest Neighbor yang di gunakan maka waktu Komputasi sistem akan semakin bertambah. Waktu paling rendah pada nilai $k=1$. Dapat di simpulkan bahwa hasil pengujian skenario 3 dengan nilai $k=1$ menghasilkan akurasi 83,33% dengan waktu komputasi selama 0,19 detik.

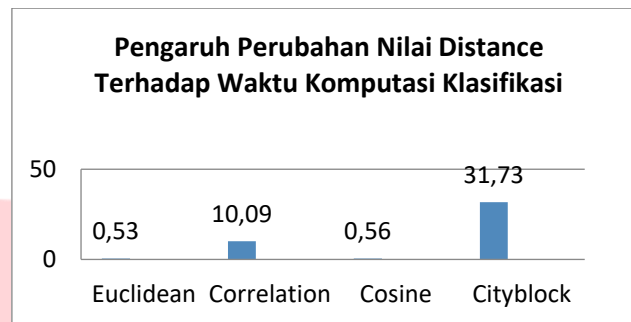
4.4 Hasil Pengujian Skenario 4

Pada pengujian skenario 4, dilakukan analisis pengaruh distance dalam metode K -Nearest Neighbor terhadap akurasi dan waktu komputasi sistem. Akurasi yang dihitung pada pengujian kali ini adalah akurasi sistem dalam proses klasifikasi domba berdasarkan database. Akurasi yang dihitung pada pengujian ini menggunakan *ratio* 0.3 dan ukuran matriks ciri dimensi fraktal $s = [2\ 4\ 8\ 16\ 32\ 64]$ dan k K - Nearest dengan nilai 1. *Distance* yang di gunakan yaitu *Euclidean*, *Correlation*, *Cosine*, dan *Cityblock*. Penguji skenario dilakukan pada 234 citra latih dan 78 citra uji.



Gambar 10. Pengaruh Perubahan Nilai *Distance* Terhadap Akurasi Klasifikasi

Berdasarkan Gambar 10 dapat dilihat dari 4 *distance* yang digunakan nilai akurasi paling baik atau bagus terdapat pada *distance Euclidean* dan *Cityblock* dengan akurasi sebesar 83,33% . hal ini di karenakan 4 aturan jarak ini memiliki persamaan yang berbeda beda sehingga menghasilkan jarak terdekat yang berbeda beda. Selanjutnya hasil pengujian pengaruh perubahan nilai *distance* terhadap waktu komputasi direpresentasikan oleh Gambar 11 berikut ini :



Gambar 11. Pengaruh Perubahan Nilai Distance Terhadap Akurasi Klasifikasi

5. Penutup

5.1 Kesimpulan

Berdasarkan hasil analisis skenario pengujian yang di lakukan pada sistem estimasi bobot karkas domba menggunakan metode fraktal dan klasifikasi *K-Nearest Neighbor*, maka didapatkan kesimpulan sebagai berikut:

1. Pengimplementasian sistem dapat menghasilkan tingkat akurasi estimasi yang cukup tinggi yakni mencapai 83,33% dengan waktu komputasi rata-rata 0,49 detik.
2. Tingkat akurasi dan waktu Komputasi optimal diperoleh dengan nilai dari masing-masing parameter yang digunakan adalah sebagai berikut: *Ratio* berukuran 0.3, jumlah matriks ciri dimensi fraktal adalah 6, parameter *s* sama dengan [2 4 8 16 32 64], nilai *k* sama dengan 1, dan perhitungan *distance* (jarak) *euclidean distance* pada klasifikasi *K-Nearest Neighbor*.
3. Nilai *ratio* pada *pre-processing* akan mempengaruhi hasil segmentasi dan akurasi estimasi sistem.
4. Ukuran maktriks ciri atau jumlah dimensi *Fractal* yang dihitung mempengaruhi akurasi dan waktu komputasi sistem.
5. Nilai *k* yang digunakan dalam *K-Nearest Neighbor* mempengaruhi akurasi klasifikasi dan waktu komputasi sisitem. Namun di peroleh nilai akurasi paling baik adalah ketika *k* bernilai sama dengan 1, dengan demikian ketika nilai *k* kecil maka hanya tetangga yang memiliki kedekatan data terbaik saja yang digunakan untuk proses klasifikasi.
6. Jenis aturan jarak atau *distance* yang digunakan dalam *K-Nearest Neighbor* mempengaruhi akurasi klasifikasi dan waktu komputasi sisitem. Namun di peroleh nilai akurasi paling baik digunakan adalahn jenis *euclidean*.

5.2 Saran

Sistem estimasi bobot karkas domba pada tugas akhir ini dapat dikembangkan dengan tujuan meningkatkan performansi kinerja sistem. Berikut adalah beberapa saran dari penulis untuk pengembangan sistem selanjutnya:

1. Untuk mendapatkan hasil akurasi sistem yang optimal, teknik pengambilan gambar atau citra harus di perhatikan . seperti intesitas cahaya, perbedaan warna objek dan background dan jarak ambil pada objek. Memperbanyak citra latih juga dapat dilakukan untuk meningkatkan akurasi.
2. Tahapan selanjutnya menggunakan algoritma yang lebih baik dan paramater yang lebih akurat dengan tujuan menghasilkan segmentasi yang bagus sehingga dapat meningkatkan akurasi sistem.
3. Sistem selanjutnya dapat di gunakan secara *real time* menggunakan *gadget* yang *compatible*.

Daftar Pustaka

- [1] R. Blackhurst, "Pemasukan Dan Pengawasan Peredaran Karkas, Daging, dan atau Jeroan dari Luar Negri," 2009.
- [2] M. F. A, Dudi, and I. A. Siwi, "Penyimpangan Bobot Badan Dugaan Menggunakan Rumus Winter Dan Rumus Arjodarmoko Terhadap Bobot Badan Aktual Sapi Pasundan Di Kabupaten Garut," pp. 1–13, Student e-journal : vol. 6, no. 1 Bandung, 2017.
- [3] R. D. Kusumanto and A. N. Tompunu, "Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB," Seminar Nasional Teknologi Informasi dan Komunikasi Terapan, pp. 1–7, Palembang, 2011.
- [4] Y. Tao., E. C. M. Lam., and Y. Y., " *Extr. Fractal Feature Pattern Recognit*", p. IEEE : 527-530., 2000.
- [5] D. Putra, *Pengolahan Citra Digital*. Yogyakarta: ANDI, 2009.
- [6] S. Muliawati, "Estimasi Berat Karkas Sapi Berdasarkan Segmentasi Mean Shift Dengan Klasifikasi Support Vector Machine Linear Beef Carcass Estimation Based on Mean Shift Segmentation With," E-Proceeding of Engineering : vol.3, no.2, Bandung, 2016.
- [7] E. Budianita, J. Jasril, and L. Handayani, "Implementasi Pengolahan Citra dan Klasifikasi K-Nearest Neighbour Untuk Membangun Aplikasi Pembeda Daging Sapi dan Babi," J. Sains dan Teknol. Ind., Pekan Baru, 2015.

