Identifikasi *Neckband* pada Sapi Menggunakan Algoritma *Random Forest*, Regresi Logistik dan KNN

Bobby Williams K. Hara (G64190061), Matthew Martianus Henry (G64190072), Irgi Muttaqin Fahrezi (G64190103), Risda Awalia (G64190106), Feliany Dwisusanta (G64190113)

Dept. Ilmu Komputer, Institut Pertanian Bogor, Indonesia
bobbywilliams@apps.ipb.ac.id[1], matthenry@apps.ipb.ac.id[2], irgimuttaqinfahrezi@apps.ipb.ac.id[3], awaliarisda26risda@apps.ipb.ac.id[4], felianydwisusanta@apps.ipb.ac.id[5]

Abstrak

Peternakan adalah kegiatan mengembangbiakkan dan membudidayakan hewan ternak untuk mendapatkan manfaat dan hasil. Salah satu hewan ternak adalah sapi perah yang merupakan hewan ternak penghasil susu. Sapi perah pada umumnya diberikan *neckband* khusus sebagai tanda pengenal. *Neckband* yang dikenakan setiap sapi memiliki pola yang berbeda. Identifikasi untuk membedakan setiap *neckband* pada sapi dapat dilakukan dengan algoritma *random forest, K-nearest neighbors* (KNN), dan regresi logistik. Sebelum dilakukan klasifikasi, data citra dilakukan proses augmentasi untuk meningkatkan keragaman pada data. Fitur yang digunakan sebagai input pada citra didapatkan dari *histogram of oriented gradients* (HOG). Klasifikasi dengan model *random forest* menghasilkan akurasi sebesar 91.86%, regresi logistik sebesar 84.79% dan KNN sebesar 88.86%. Dari ketiga algoritma yang digunakan, didapatkan *random forest* sebagai model terbaik untuk membedakan *neckband* yang dikenakan pada sapi.

Kata Kunci**:** *histogram of oriented gradients*, K-*nearest neighbors*, *neckband*, *random forest,* regresi logistik, sapi

PENDAHULUAN

Latar Belakang

Peternakan adalah kegiatan mengembangbiakkan dan membudidayakan hewan ternak untuk mendapatkan manfaat dan hasil dari kegiatan tersebut. Pengertian peternakan tidak terbatas pada pemeliharaan saja, terdapat perbedaan antara memelihara dan peternakan, yakni terdapat pada tujuan yang ditetapkan. Peternakan bertujuan mencari keuntungan dengan menerapkan prinsip manajemen pada faktor-faktor produksi yang telah dikombinasikan secara optimal. Bidang peternakan dapat dibagi atas dua golongan berdasarkan ukuran hewan ternak, yaitu peternakan hewan besar seperti sapi, kerbau dan kuda dan peternakan hewan kecil seperti ayam, kelinci dan lain-lain. Pada penelitian ini, obyek dari dataset merupakan hewan yang biasanya dipelihara oleh peternakan.

Sapi perah merupakan salah satu contoh hewan ternak penghasil susu. Bangsa sapi perah yang memiliki tingkat produksi susu paling tinggi adalah sapi Fries Holland (FH). Blakely dan Bade (1994) menyatakan bahwa produksi susu sapi perah FH di negara asalnya berkisar 6.000–7.000 liter dalam satu masa laktasi. Sudono *et al.* (2003) menyebutkan bahwa produktivitas sapi FH di Indonesia masih termasuk rendah dengan produksi susu kurang lebih 3.050 Kg/laktasi. Sapi perah merupakan binatang ternak penghasil susu utama yang mencukupi kebutuhan susu dunia dibandingkan dengan ternak penghasil susu yang lain. Pemeliharaan sapi perah selalu diarahkan pada peningkatan produksi susu karena peran pentingnya dalam mencukupi kebutuhan susu dunia.

Undang-undang yang diterapkan pada binatang ternak di wilayah Indonesia diatur dalam Undang-Undang Nomor 18 Tahun 2009 tentang Peternakan Dan Kesehatan Hewan.

Undang-undang ini wajib ditaati oleh para pemangku kepentingan *(stakeholders)*. Peran pemerintah dalam mengatur peternakan khususnya ternak sapi perah di Indonesia adalah memberikan arah melalui pembuatan kebijakan yang bertujuan meningkatkan produktivitas peternak sapi perah dalam melakukan kegiatan peternakannya (Hasibuan 2016). Peternak sapi perah harus ikut berperan dalam menjalankan peternakan sesuai dengan kebijakan yang telah ditentukan oleh Pemerintah. Selain itu, Industri Pengolahan Susu ikut berperan dalam mendayagunakan hasil produksi susu dari peternak sapi perah kepada masyarakat.

History of Oriented Gradients (HOG) merupakan sebuah metode yang digunakan pada *image processing* yang bertujuan untuk melakukan deteksi objek (Anggraeny *et al.* 2020). Teknik ini menghitung nilai gradien dalam daerah tertentu suatu citra. Karakteristik suatu citra ditunjukkan pada distribusi gradien. Karakteristik ini diperoleh dengan membagi citra ke dalam daerah kecil yang disebut sel. Pada penelitian yang dilakukan oleh Dalal dan Triggs (2005), HOG merupakan ekstraksi ciri yang dapat mendeteksi objek dengan baik dengan menggunakan SVM sebagai klasifikasinya dengan harapan tingkat akurasi dalam mendeteksi ikan dapat meningkat saat menggunakan teknik boosting salah satunya adalah algoritma Adaboost (Dalal dan Triggs 2005). Pada penelitian ini, HOG diaplikasikan pada citra untuk melakukan klasifikasi gambar.

Random Forest didefinisikan sebagai prinsip umum suatu ansambel acak dari suatu pohon keputusan (Breiman, 2001). Devella *et al.* memanfaatkan ekstraksi fitur SIFT untuk klasifikasi motif songket Palembang. *Keypoint* didapatkan dengan memanfaatkan ekstraksi SIFT dan *keypoint* bermanfaat dalam mempengaruhi *noise* pada citra. Klasifikasi dilakukan dengan algoritma *Random Forest* dan menghasilkan akurasi yang sangat baik. Pada penelitian ini, algoritma *random forest* diimplementasikan ke dataset yang telah ada untuk dilakukan tahapan klasifikasi. Klasifikasi dengan menggunakan algoritma ini memiliki akurasi yang sangat baik. Berikut tahapan konstruksi *random forest* yang dapat dilakukan:

1. Menggambar sampel *bootstrap* n-tree dari data.
2. Menumbuhkan pohon untuk setiap kumpulan data *bootstrap*. Di setiap simpul pohon, pilih variabel entri secara acak untuk dipisahkan, kemudian tumbuhkan pohon sehingga setiap node terminal memiliki tidak kurang dari kasus ukuran node.
3. Informasi agregat dari pohon n-tree untuk prediksi data baru.
4. Menghitung tingkat kesalahan *out of bag* (OOB) menggunakan data yang bukan sampel *bootstrap*.

K-Nearest Neighbor (KNN) merupakan suatu metode klasifikasi yang sederhana dan efektif dan melakukan klasifikasi sesuai dengan banyaknya kedekatan jarak yang mayoritas terhadap data yang diklasifikasi (Santoso, 2007). Klasifikasi dengan menggunakan metode KNN menghasilkan akurasi yang akurat jika *data training* yang digunakan jumlahnya banyak. Thepade *et al.* menggunakan ekstraksi fitur GLCM dan mengkombinasikannya dengan *wavelet transform* serta menggunakan KNN dan SVM sebagai *classifier* untuk mengidentifikasi COVID-19 pada citra X-Ray paru (Thepade *et al.* 2020). Pe yg.nggunaan KNN pada penelitian ini menghasilkan nilai akurasi paling tinggi, yakni sebesar 92,6%.

Regresi logistik adalah bentuk regresi yang digunakan untuk memodelkan hubungan antara variabel dependen dan variabel independen. Variabel dependen adalah data yang berukuran dikotomi/biner, misalnya ya atau tidak, sukses atau gagal, dsb, sementara variabel independen adalah data yang berjenis nominal, interval, rasio, dsb. Regresi logistik dapat digunakan untuk memprediksi variabel dependen oleh sebuah atau beberapa variabel dependen, menentukan persentase varians dalam variabel dependen yang dapat dijelaskan oleh variabel independen, dan menentukan peringkat kepentingan relatif variabel independen terhadap variabel dependen (Alfian 2016). Chen *et al.* dalam karya tulis *"Vehicle Logo Recognition by Spatial-SHIFT Combined with Logistic Regression"* mengombinasikan SIFT dengan Logistic Regression (Chen *et al.*). Kelemahan dari regresi logistik sesuai dengan

penelitian yang telah disebutkan sebelumnya adalah regresi logistik rentan terhadap *underfitting* dan memiliki akurasi yang rendah.

Rumusan Masalah

1. Apa algoritma terbaik untuk mengidentifikasi *neckband* pada sapi?

Tujuan Penelitian

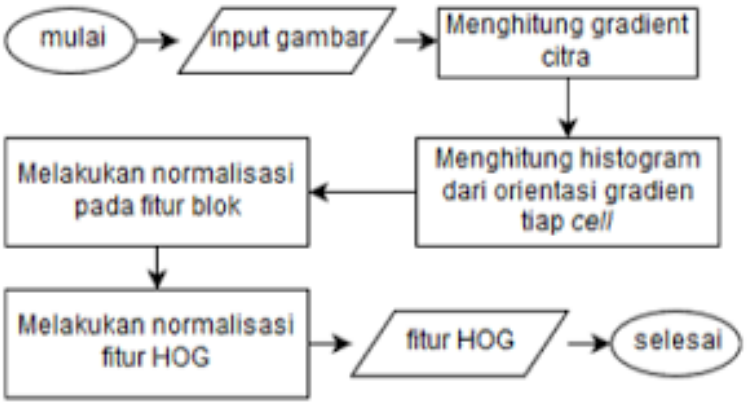
1. Mengetahui algoritma terbaik dalam mengidentifikasi *neckband* sapi.

TINJAUAN PUSTAKA

Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients (HOG) merupakan salah satu teknik pengambilan fitur yang bertujuan untuk mengambil informasi penting dari sebuah citra. Metode ini bekerja dengan mengevaluasi histogram lokal yang sudah ternormalisasi secara baik dari distribusi gradien citra dalam *grid* yang padat. Metode HOG banyak digunakan pada *computer vision*. HOG adalah deskriptor berbasis *window* yang mendeteksi titik *interest*. Metode ini menghitung nilai gradien dalam daerah tertentu pada suatu citra. Setiap citra memiliki karakteristik yang ditunjukkan oleh distribusi gradien yang diperoleh dengan membagi citra ke dalam daerah kecil yang disebut *cell*. Setiap *cell* disusun dari sebuah histogram dari sebuah gradien. Kombinasi dari histogram ini dijadikan sebagai deskriptor yang mewakili sebuah objek. (N & B, 2005).

Proses awal pada metode HOG adalah dengan konversi citra RGB (*red, green, blue*) menjadi *grayscale* yang kemudian dilanjutkan dengan menghitung nilai gradien setiap piksel. Setelah nilai gradien didapatkan, proses selanjutnya adalah menentukan jumlah *bin* orientasi yang akan digunakan dalam pembuatan histogram. Proses ini disebut *spatial orientation binning*. Namun sebelumnya pada proses perhitungan gradien, gambar pelatihan dibagi menjadi beberapa *cell* dan dikelompokkan menjadi ukuran lebih besar yang disebut *block*. Pada proses normalisasi *block* digunakan perhitungan geometri R-HOG. Proses ini dilakukan karena terdapat *block* yang saling tumpang tindih. Algoritma pada proses HOG dapat dilihat pada gambar berikut :



Gambar 1 Prosedur algoritma HOG

Dari Gambar 1, tahap awal dari metode HOG adalah menghitung nilai gradien citra dihitung menggunakan

$$|G| = \sqrt{I_x^2 + I_y^2}$$

Dimana I adalah citra gray level. *Ix* merupakan matrik terhadap sumbu-x dan *Iy* merupakan matrik terhadap sumbu-y. *Ix* dan *Iy* dapat dihitung dengan

$$I_x = I * S_x,\ I_y = I * S_y$$

Sy dan *Sx* didapatkan dari metode untuk mencari nilai gradien yaitu dengan metode Sobel :

$$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

$$S_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Gambar 2 Kernel Sobel

Masing-masing kernel dihitung dengan cara konvolusi. Selanjutnya gradien ditransformasi ke dalam koordinat sumbu dengan sudut diantara 0-180 derajat yang disebut orientasi gradien. Fitur *block* dinormalisasi untuk mengurangi efek perubahan kecerahan objek pada satu *block*. Orientasi gradien (θ) dapat dihitung dengan persamaan :

$$\theta = arctan\left(\frac{lx}{ly}\right)$$

Nilai normalisasi fitur blok didapat dari persamaan :

$$b = \frac{b}{\sqrt{b^{2+e}}}$$

b = nilai blok fitur
e = 2,71828

Nilai normalisasi tiap *block* digabungkan menjadi satu vektor menjadi fitur vektor HOG. Kemudian fitur vektor HOG dilakukan normalisasi. Normalisasi dilakukan melalui persamaan :

$$h = \frac{h}{\sqrt{||h||^{2+3}}}:$$

h = nilai fitur HOG
e = 2,71828 (konstanta)

Random Forest

Random forests adalah suatu metode klasifikasi yang terdiri dari gabungan pohon klasifikasi (CART) yang saling independen. Prediksi klasifikasi diperoleh melalui proses voting (jumlah terbanyak) dari pohon-pohon klasifikasi yang terbentuk. *Random forests* merupakan pengembangan dari metode *ensemble* yang pertama kali dikembangkan oleh Leo Breiman (2001) dan digunakan untuk meningkatkan ketepatan klasifikasi. Dalam proses *bagging* digunakan *resampling bootstrap* untuk membangkitkan pohon klasifikasi dengan banyak versi yang kemudian dikombinasikan untuk memperoleh prediksi akhir. *Random forest* menggunakan proses pengacakan untuk membentuk pohon klasifikasi. Proses ini dilakukan tidak hanya dilakukan untuk data sampel saja melainkan juga pada pengambilan variabel prediktor. Proses ini akan menghasilkan kumpulan pohon klasifikasi dengan ukuran dan bentuk yang berbeda-beda. Hasil yang diharapkan adalah suatu kumpulan pohon klasifikasi

yang memiliki korelasi kecil antar pohon. Korelasi yang kecil akan menurunkan hasil kesalahan prediksi dari *random forest* (Breiman, 2001).

Random forest merupakan suatu metode klasifikasi yang berisi koleksi dari pohon klasifikasi. Misalkan $*h(x, \Theta_k), k = 1, \ldots +$ dimana $*\Theta_k+$ merupakan vektor random yang iid (*independent identically distributed*) dan tiap pohon memilih kelas yang paling banyak dari data (*majority vote*). Misalkan suatu *ensemble* $h_1(x), h_2(x), \ldots, h_K(x)$ dengan data latih dipilih secara acak dari distribusi vektor random Y dan X, fungsi margin (mg(X, Y)) dari *random forest* didefinisikan sebagai berikut :

$$mg(X,Y) = \frac{\sum_{k=1}^{K} I(h_k(X)=Y)}{K} - max\,(j \neq y)\left[\frac{\sum_{k=1}^{K} I(h_k(X)=J)}{K}\right]$$

dimana I adalah fungsi indikasi dan K adalah banyaknya pohon. Fungsi margin digunakan untuk mengukur tingkat banyaknya jumlah vote pada X dan Y rata-rata *vote* dari kelas yang lain.

Strength (kekuatan) adalah rata-rata (ekspektasi) ukuran kekuatan akurasi pohon tunggal. Nilai s yang semakin besar menunjukkan bahwa akurasi prediksinya semakin baik. Nilai s didefinisikan sebagai berikut (Breiman, 2001) :

$$S = E_{x,y}\, mg(X,Y)$$

Rata-rata korelasi ($\bar{\varrho}$) antar pasangan dugaan dari dua pohon tunggal dalam *random forest* didefinisikan sebagai berikut (Breiman, 2001) :

$$\bar{\rho} = \frac{E_{\theta.\theta'(\rho(\theta,\theta')sd((\theta)sd(\theta'))}}{E_{\theta.\theta'(sd((\theta)sd(\theta'))}}$$

dimana $\varrho(\theta, \theta')sd(\theta)sd(\theta')$ merupakan korelasi antar pohon. Batasan besarnya kesalahan prediksi (s_{RF}) oleh *Random Forest* adalah:

$$^{\varepsilon}RF \leq \bar{\rho}\left(\frac{1-s^2}{s^2}\right)$$

Dari persamaan tersebut dapat dikatakan bahwa jika ingin memperoleh error yang kecil maka harus memiliki korelasi kecil dan memperkuat *strength*. Oleh karena itu diperlukan untuk mengubah nilai *m* dan *ntree*. Dengan memperkecil nilai *m*, maka memperkecil pula korelasi dan *strength*. Begitu pula dengan nilai *ntree*. Jika *ntree* besar berarti kesamaan data di antara tiap-tiap pohon sangat tinggi. Akan tetapi, jika pemilihan *m* dan *ntree* sangat rendah, mengartikan setiap pohon akan kehilangan beberapa informasi penting dan akan menaikkan error. Sehingga pemilihan *m* dan *ntree* sangat berpengaruh dalam *Random Forest*.

K-Nearest Neighbor (KNN)

K-Nearest Neighbor (kNN) termasuk kelompok *instance-based learning*. Algoritma ini merupakan salah satu teknik *lazy learning*. KNN dilakukan dengan mencari kelompok *k* objek dalam data *training* yang paling dekat (mirip) dengan objek pada data baru atau data *testing* (Wu 2009).

Algoritma K-Nearest Neighbor (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek yang berdasarkan dari data pembelajaran yang jaraknya paling dekat dengan objek tersebut. KNN merupakan algoritma *supervised learning* dimana hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada algoritma KNN dimana kelas yang paling banyak muncul yang nantinya akan menjadi kelas hasil dari klasifikasi (Avelita, 2014).

Kedekatan didefinisikan dalam jarak metrik seperti jarak Euclidean seperti pada persamaan berikut :

$$Dxy = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Keterangan :

D : jarak kedekatan

x : data training

y : data testing

n : jumlah atribut individu antara 1 s.d. n

f : fungsi similitary atribut i antara kasus *X* dan kasus *Y*

i = Atribut individu antara 1 sampai dengan n.

Langkah-langkah untuk menghitung metode KNN antara lain :

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

Regresi Logistik

Regresi Logistik adalah suatu metode analisis statistika untuk mendeskripsikan hubungan antara peubah respon (*dependent variable*) yang memiliki dua kategori atau lebih dengan satu atau lebih peubah penjelas (*independent variable*) berskala kategori atau interval (Hosmer dan Lemeshow, 2000).

Regresi Logistik merupakan regresi non linear, digunakan untuk menjelaskan hubungan antara X dan Y yang bersifat tidak linear, ketidaknormalan sebaran Y, keragaman respon tidak konstan yang tidak dapat dijelaskan dengan model regresi linear biasa (Agresti 1996). Regresi logistik adalah bagian dari analisis regresi yang dapat digunakan jika variabel dependen (respon) merupakan variabel dikotomi. Variabel dikotomi biasanya hanya terdiri atas dua nilai, yang mewakili kemunculan atau tidak adanya suatu kejadian yang biasanya diberi angka 0 atau 1 (Nirwana 2015).

Secara umum, persamaan regresi logistik untuk *k* variabel prediktor (Nirwana 2015) terdapat pada persamaan :

$$ln[odds(T/X1,\ X2,\ .\ .\ Xk)]\ =\ \beta0\ +\ \beta1X1\ +\ \beta2X2\ +.\ .\ +\ \beta kXk$$

Regresi logistik akan membentuk variabel prediktor atau respon ($ln\ (P/(1-P))$ yang merupakan kombinasi linier dari variabel independen. Nilai variabel prediktor ini kemudian ditransformasikan menjadi probabilitas dengan fungsi logit. Model regresi linear sederhana terdapat pada persamaan:

$$Y_i\ =\ \beta_0\ +\ \beta_1 X_i\ +\ e_i$$

di mana Y_i merupakan variabel respon, β_0 serta β_1 merupakan parameter, e_i merupakan galat ke *i*, untuk *i = 1, 2, ..., n*. Apabila persamaan merupakan model regresi yang tidak memiliki intersep maka persamaan tersebut terdapat pada persamaan :

$$Y_i\ =\ \beta_1 X_i\ +\ s_i$$

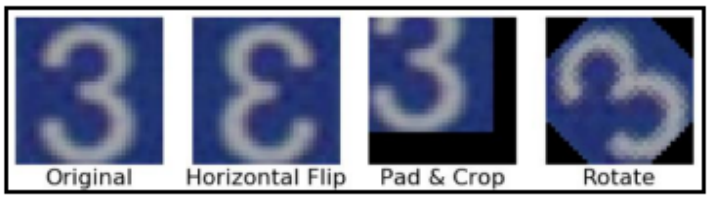
di mana Y_i merupakan variabel respon, β_0 serta β_1 merupakan parameter, e_i merupakan galat ke *i*, untuk *i = 1, 2, ..., n*. Bila diambil pengamatan sebanyak *n* maka persamaan ini terdapat pada persamaan :

$$Y\ =\ \beta_1 X\ +\ e$$

Variabel respon dalam persamaan regresi tidak hanya dipengaruhi oleh variabel bebas yang bersifat kuantitatif saja (umur, pendapatan, harga dan sebagainya), tetapi seringkali juga dipengaruhi oleh variabel yang bersifat kualitatif (seperti jenis kelamin, musim, warna dan sebagainya).

Augmentasi Data

Augmentasi data adalah strategi yang memungkinkan praktisi untuk secara signifikan meningkatkan keragaman data yang tersedia untuk model pelatihan, tanpa benar-benar mengumpulkan data baru. Teknik augmentasi data seperti cropping, padding, dan flipping horizontal umumnya digunakan untuk melatih jaringan neural besar. Namun, sebagian besar pendekatan yang digunakan dalam pelatihan jaringan neural hanya menggunakan tipe augmentasi dasar. Sementara arsitektur jaringan neural telah diselidiki secara mendalam. Contoh augmentasi data terhadap gambar dapat dilihat pada gambar 2 [12].



Gambar 3 Citra original dan hasil augmentasi

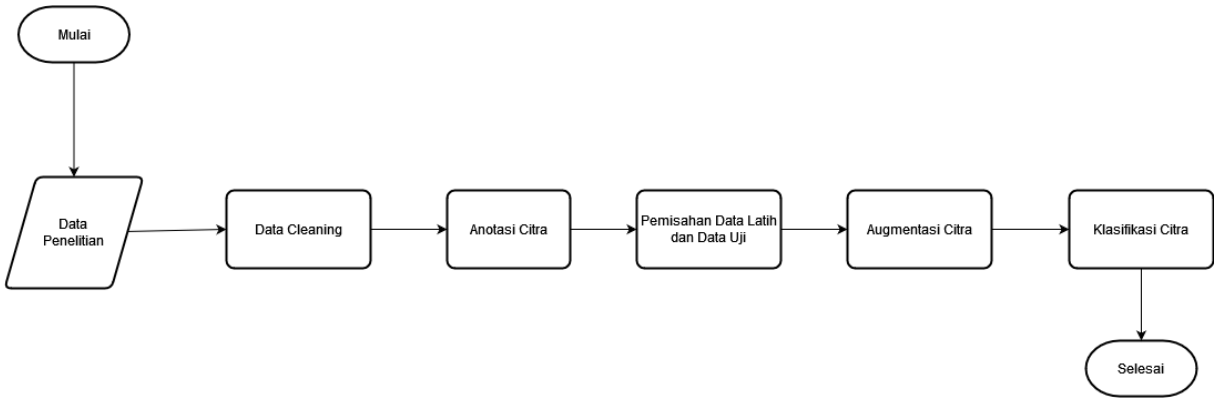
Augmentasi dapat meningkatkan akurasi dari model CNN yang dilatih karena dengan augmentasi model mendapatkan data-data tambahan yang dapat berguna untuk membuat model yang dapat melakukan generalisasi dengan lebih baik. Augmentasi yang dilakukan pada penelitian ini adalah membalikan gambar secara horizontal, melakukan zoom-in secara acak,

dengan maksimal zoom sebesar 50% dari besar gambar, dan juga melakukan rotasi gambar secara acak dengan derajat maksimal 90◦ . Salah satu jenis augmentasi yang umum dilakukan adalah dengan melakukan perputaran gambar dengan besar tertentu, contoh dari penggunaan augmentasi ini dapat dilihat pada Gambar 4.

LINGKUNGAN PENGEMBANGAN

1. Hardware
    - Processor: Intel(R) Core(TM) i5-4210H CPU @ 1.70 GHz
    - Memory : 8 GB DDR4
    - Storage: 256 GB SSD
2. Software
    - Text Editor : Jupyter Notebook
    - Operating System : Windows 10
    - Bahasa Pemrograman : Python 3.10

METODE



Gambar 4 Diagram alur pengerjaan proyek

Data Penelitian

Data penelitian yang digunakan berupa citra sapi dengan *neckband* di lehernya. Data tersebut terpisah pada delapan folder dengan masing masing folder menyatakan kelas dari *neckband* sapi tersebut. Berikut rincian kelas beserta jumlah citra yang terdapat di dalamnya:

Kelas	Jumlah Citra
1	330
2	299
3	316
4	202
5	200
6	200
7	200
8	201

Data Cleaning

Tahapan ini dilakukan untuk memisahkan citra yang dapat mengganggu proses augmentasi dan klasifikasi. Pemisahan ini dilakukan secara manual dengan melihat setiap citra pada folder kelasnya dan menghapus setiap citra yang tidak jelas atau dapat mengganggu proses augmentasi. Citra tersebut berupa citra sapi dengan *neckband* yang tak terlihat dan citra sapi yang tidak terlihat sama sekali akibat cahaya yang masuk terlalu banyak saat akuisisi gambar.

Anotasi Citra

Tahapan ini dilakukan untuk memberikan label pada citra sehingga setiap citra memiliki kelas tertentu yang dapat diklasifikasi oleh *classifier*. Anotasi dilakukan pada data citra yang telah melalui proses *data cleaning*.

Data citra pada setiap folder kelas yang telah melalui proses *data cleaning* diberi label kelas tersebut dan datanya disimpan dalam bentuk CSV. Pada CSV terdapat 2 kolom yaitu kolom *image* yang menyimpan citra dalam format bitmap dan kolom *class* yang menyatakan kelas citra tersebut.

Hasil CSV dari setiap kelas kemudian digabungkan menjadi sebuah dataset yang akan digunakan pada proses klasifikasi. Dataset CSV tersebut kemudian digabung bersama seluruh citra dari setiap folder kelas dalam sebuah folder baru.

Pemisahan Data Menjadi Data Latih dan Data Uji

Data citra yang telah dianotasi dipisahkan menjadi data latih dan data uji. Proporsi pemisahan yang digunakan adalah 70% untuk data latih dan 30% untuk data uji.

Augmentasi Citra

Tahapan ini dilakukan untuk memperbanyak gambar pada data latih dan mengenalkan keragaman pada model klasifikasi. Pengenalan keragaman tersebut diharapkan dapat meningkatkan akurasi model klasifikasi.

Pengimplementasian augmentasi citra dilakukan dengan Keras ImageDataGenerator dari *library* TensorFlow. Class ImageDataGenerator menyediakan beberapa parameter yang dapat digunakan untuk melakukan proses augmentasi, seperti *rotation_range*, *brightness_range*, *width_shift_range*, *height_shift_range*, *shear_range*. Nilai dari *rotation_range* adalah 90, *brightness_range* adalah [0.2, 1.0], *width_shift_range* dan *height_shift* adalah 0.1 dan *shear_range* adalah 0.1 .

***Histogram of Oriented Gradients* (HOG)**

Tahapan ini dilakukan untuk mengubah citra menjadi sebuah matriks HOG yang akan menjadi masukan untuk model klasifikasi.

Klasifikasi Citra

Tahapan ini merupakan tahap klasifikasi citra data latih yang telah diperbanyak melalui proses augmentasi. Model yang digunakan adalah model regresi logistik, *random forest* dan KNN. Model yang telah dibuat tersebut kemudian digunakan untuk mengklasifikasi setiap citra pada data uji dan menentukan kelasnya.

HASIL DAN PEMBAHASAN

Data Cleaning

Pada data penelitian yang digunakan, terdapat banyak citra yang bersifat *noise*, yaitu citra sapi dengan pencahayaan yang terlalu banyak dan citra sapi dengan *neckband* yang tidak

terlihat. Citra di kiri bawah merupakan contoh sapi dengan *neckband* yang tidak terlihat dan citra di sebelah kanan bawah menunjukkan cahaya yang terlalu banyak sehingga citra sapi menjadi tidak jelas.

Gambar 5 Sapi dengan neckband yang tidak terlihat (kiri) dan cahaya terlalu banyak (kanan)

Penghapusan citra tersebut menyebabkan data yang digunakan menjadi hanya berjumlah 1554 citra dari sebelumnya 1948 citra.

Anotasi Citra

Data citra yang dilakukan proses anotasi memiliki format CSV dengan data yang ditampung adalah nama gambar dan kelasnya. Dataset ini memiliki 1554 baris dan 2 kolom. Kolom pertama menyatakan nama citra dalam format bitmap dan kolom kedua menyatakan kelas dari citra tersebut.

	image	class
0	WA01_2015_10_1_16_22_15_253889_id1.bmp	1
1	WA01_2015_10_1_16_22_15_253889_id10.bmp	1
2	WA01_2015_10_1_16_22_15_253889_id11.bmp	1
3	WA01_2015_10_1_16_22_15_253889_id2.bmp	1
4	WA01_2015_10_1_16_22_15_253889_id3.bmp	1
...	...	...
1549	WA08_2016_10_2_8_51_15_737651_id6.bmp	8
1550	WA08_2016_10_4_21_34_21_735098_id3.bmp	8
1551	WA08_2016_10_8_7_11_55_440456_id1.bmp	8
1552	WA08_2016_10_8_7_11_55_440456_id2.bmp	8
1553	WA08_2016_10_9_11_11_18_331387_id7.bmp	8

1554 rows × 2 columns

Gambar 6 Dataset CSV yang telah dianotasi

Pemisahan Data Latih dan Data

Sebelum dilakukan augmentasi data pada data latih, data keseluruhan dibagi menjadi data latih dan data uji. Perbandingan data latih dan data uji adalah 70 banding 30, sehingga jumlah data latih adalah 1087 dan data uji 467.

	image	class
796	WA01_2015_11_21_13_24_5_636357_id3.bmp	4
197	WA01_2015_11_20_15_22_12_912941_id9.bmp	1
83	WA01_2015_10_3_7_55_31_120821_id6.bmp	1
889	WA01_2015_11_30_8_45_7_813862_id7.bmp	4
891	WA01_2015_11_30_8_45_7_813862_id9.bmp	4
...	...	...
29	WA01_2015_10_2_7_8_13_295678_id11.bmp	1
1097	WA01_2015_11_20_14_7_28_674052_id6.bmp	6
1027	WA08_2016_10_14_9_38_52_958190_id6.bmp	5
759	WA01_2015_10_1_16_10_39_129950_id6.bmp	4
367	WA08_2016_11_14_15_52_57_96778_id8.bmp	2

1087 rows × 2 columns

	image	class
1133	WA01_2015_11_21_8_52_9_829348_id5.bmp	6
673	WA08_2016_11_14_9_12_51_529193_id7.bmp	3
712	WA08_2016_11_7_20_1_40_761535_id1.bmp	3
916	WA01_2015_10_1_16_1_38_222464_id1.bmp	5
433	WA08_2017_3_10_10_44_27_634856_id4.bmp	2
...	...	...
649	WA08_2016_11_12_9_50_49_473954_id6.bmp	3
928	WA01_2015_10_2_7_38_13_765178_id11.bmp	5
738	WA08_2016_11_8_9_51_35_887216_id3.bmp	3
63	WA01_2015_10_3_2_4_48_69792_id4.bmp	1
550	WA08_2016_10_17_15_51_7_838207_id5.bmp	3

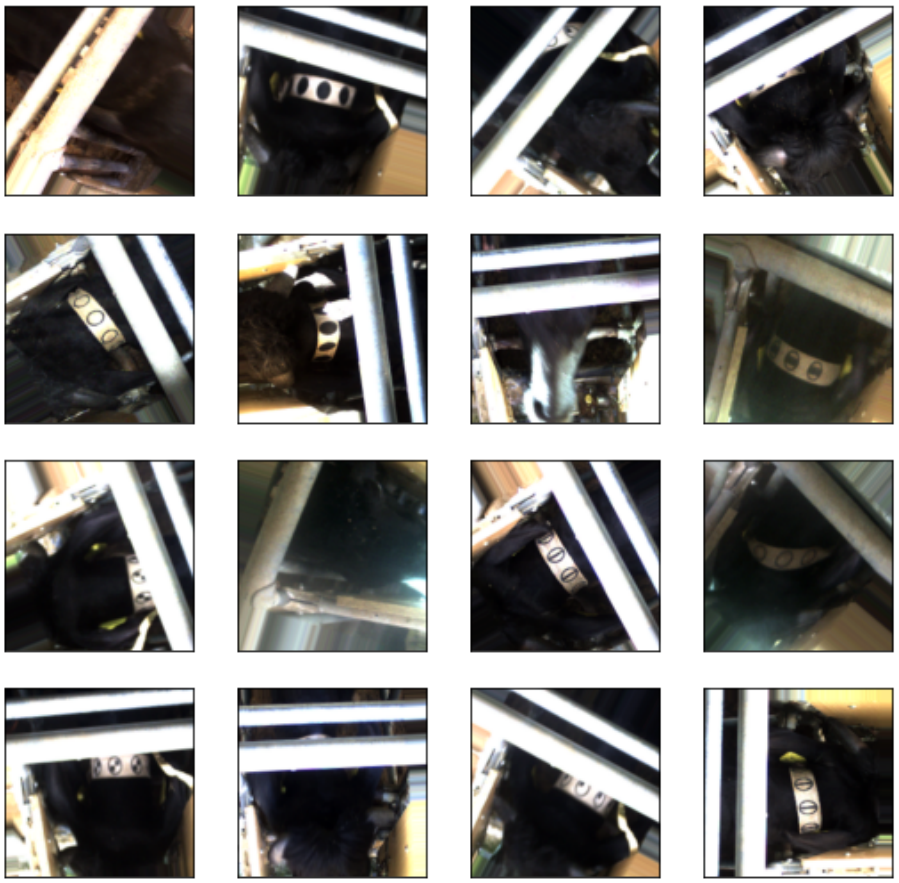
467 rows × 2 columns

Gambar 7 Data latih (kiri) dan data uji (kanan)

Augmentasi

Proses augmentasi yang digunakan termasuk jenis *offline augmentation,* yaitu citra hasil augmentasi disimpan terlebih dahulu dalam suatu folder pada *local disk* sebelum digunakan dalam training. Dalam melakukan augmentasi citra, terdapat beberapa proses yang dilakukan, seperti rotasi citra, peningkatan *brightness* dan *shear* citra serta translasi citra.

Setelah dilakukan augmentasi citra, gambar menjadi lebih bervariasi dan lebih banyak. Contoh beberapa citra yang telah dilakukan augmentasi dapat dilihat pada Gambar 8.



Gambar 8 Data citra hasil augmentasi

Data latih yang telah dilakukan augmentasi disimpan terlebih dahulu di folder sesuai dengan nama kelas yang bersesuaian. Setelah itu, dilakukan penggabungan tiap path images tiap folder sesuai class sehingga menjadi sebuah *data frame* dengan label kelas yang sesuai. Jumlah baris pada data latih yang telah diaugmentasi adalah 4233 baris seperti pada Gambar 9.

	image	class
0	augmented-images/class1\class1_0_1016.bmp	1
1	augmented-images/class1\class1_0_1063.bmp	1
2	augmented-images/class1\class1_0_1075.bmp	1
3	augmented-images/class1\class1_0_1081.bmp	1
4	augmented-images/class1\class1_0_1087.bmp	1
...	...	...
4228	augmented-images/class8\class8_0_9848.bmp	8
4229	augmented-images/class8\class8_0_988.bmp	8
4230	augmented-images/class8\class8_0_9887.bmp	8
4231	augmented-images/class8\class8_0_9904.bmp	8
4232	augmented-images/class8\class8_0_9969.bmp	8

4233 rows × 2 columns

Gambar 9. Banyaknya data setelah augmentasi

Data frame dari data latih selanjutnya digabung dengan *data frame* dari data yang telah dilakukan augmentasi, sehingga menghasilkan data latih dengan jumlah data yang lebih banyak yaitu sebanyak 5320 baris dan lebih bervariasi seperti pada Gambar 10.

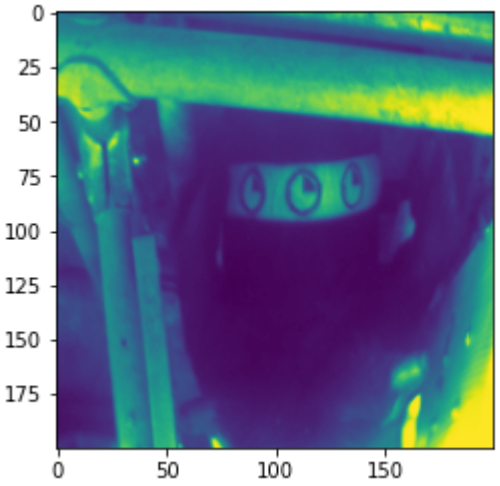
	image	class
0	WA01_2015_11_21_13_24_5_636357_id3.bmp	4
1	WA01_2015_11_20_15_22_12_912941_id9.bmp	1
2	WA01_2015_10_3_7_55_31_120821_id6.bmp	1
3	WA01_2015_11_30_8_45_7_813862_id7.bmp	4
4	WA01_2015_11_30_8_45_7_813862_id9.bmp	4
...	...	...
5315	augmented-images/class8\class8_0_9848.bmp	8
5316	augmented-images/class8\class8_0_988.bmp	8
5317	augmented-images/class8\class8_0_9887.bmp	8
5318	augmented-images/class8\class8_0_9904.bmp	8
5319	augmented-images/class8\class8_0_9969.bmp	8

5320 rows × 2 columns

Gambar 10. Banyaknya data setelah penggabungan hasil augmentasi dengan data latih sebelumnya

***Histogram of Oriented Gradients* (HOG)**

Data latih yang telah diperbanyak melalui proses augmentasi beserta data uji kemudian diubah menjadi gambar *grayscale*.



Gambar 11. Data citra format *grayscale*

Setelah citra dibuat dalam format *grayscale*, dilakukan perhitungan nilai *histogram of oriented gradients* (HOG) dari citra *grayscale* yang digunakan. Nilai tersebut disimpan dalam sebuah matriks yang berisi kumpulan *array* nilai HOG. Kolom dari matriks merupakan lokasi *feature descriptor* dan nilai dari kolom merupakan nilai *feature descriptor* HOG pada titik tersebut. Matriks tersebut memiliki 1152 kolom dan akan digunakan sebagai fitur untuk model klasifikasi gambar. Untuk matriks citra latih, matriks tersebut kemudian disatukan dengan label kelasnya. Berikut adalah *data frame* HOG dari data latih yang digunakan dengan banyak fitur berjumlah 1153. Data latih yang digunakan sebanyak 5320 dan data uji sebanyak 467.

	0	1	2	3	4	5	6	7	8	9
0	0.117995	0.050710	0.443747	0.533081	0.533081	0.459177	0.071262	0.048193	0.117805	0.023302
1	0.320275	0.159452	0.457383	0.457383	0.457383	0.426551	0.128114	0.214574	0.372708	0.028981
2	0.122302	0.006824	0.026420	0.649524	0.649524	0.094755	0.123774	0.340935	0.038479	0.024464
3	0.085308	0.023422	0.024254	0.688499	0.688499	0.110595	0.120896	0.129142	0.090641	0.020040
4	0.073881	0.013597	0.041776	0.687841	0.687841	0.127710	0.137810	0.105166	0.091530	0.015847
...	...	...	...	...	...	...	...	...	...	...
5315	0.301210	0.003960	0.009824	0.004908	0.558460	0.724805	0.267577	0.017894	0.398433	0.398433
5316	0.143471	0.000000	0.000000	0.000000	0.227343	0.184757	0.945275	0.007173	0.234455	0.005288
5317	0.450264	0.000000	0.116232	0.005805	0.491887	0.491887	0.491887	0.240539	0.437134	0.099949
5318	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
5319	0.217332	0.012543	0.000000	0.084746	0.547407	0.547407	0.547407	0.215554	0.329428	0.023351

5320 rows × 1153 columns

Gambar 12. Matriks HOG yang menjadi input untuk model klasifikasi

Klasifikasi

Model klasifikasi yang digunakan adalah *random forest*, *logistic regression* dan KNN. Berikut merupakan hasil klasifikasi setiap modelnya ketika diterapkan pada data uji.

Tabel 1 Hasil Klasifikasi Model

Model Klasifikasi	Akurasi
Regresi Logistik	0.8479
Random Forest	0.9186
KNN	0.8886

Model *random forest* yang digunakan membentuk *decision tree* sebanyak 100 *tree*. Model ini menggunakan 100 *decision tree* yang merupakan parameter bawaan dari fungsi sklearn yang digunakan. Fitur yang digunakan untuk menentukan *best split* adalah sebanyak akar dari jumlah fitur pada *node*. Dengan menggunakan *default parameter*, model *random forest* menghasilkan akurasi sebesar 91.86% dengan *confusion matrix* seperti ditunjukkan pada gambar .

```
Random Forest Confusion Matrix :
 [[77  0  0  1  1  3  2  2]
 [ 0 63  1  0  4  6  0  2]
 [ 0  3 83  0  1  0  2  3]
 [ 0  0  0 50  0  0  0  0]
 [ 0  0  0  0 31  0  0  0]
 [ 0  1  0  0  1 48  1  4]
 [ 0  0  0  0  0  0 44  0]
 [ 0  0  0  0  0  0  0 33]]
```

Gambar 13. *Confusion matrix* untuk model *random forest*

Model selanjutnya yang digunakan adalah model regresi logistik. Model regresi logistik yang digunakan secara *default* menerapkan regularisasi L2. Regularisasi ini menyebabkan model lebih resisten terhadap *overfitting*. Model regresi logistik menghasilkan akurasi sebesar 84.79% dan *confusion matrix* yang dihasilkan seperti ditunjukkan di bawah ini.

```
Logistic Regression Confusion Matrix :
 [[77  0  0  1  0  0  3  1]
 [ 0 53  2  0  0  2  0  0]
 [ 0  7 69  3  1  1  0  2]
 [ 0  1  3 46  2  0  0  3]
 [ 0  1  4  0 28  5  0  0]
 [ 0  3  5  0  2 45  1  3]
 [ 0  2  1  0  1  1 44  1]
 [ 0  0  0  1  4  3  1 34]]
```

Gambar 14. *Confusion matrix* untuk model regresi logistik

Model lainnya yang diuji cobakan adalah model KNN. Jumlah tetangga yang digunakan dalam model adalah 5 dengan bobot jarak untuk setiap titik adalah sama (*uniform weight*). Jumlah tetangga sebanyak 5 digunakan karena faktor waktu komputasional. Jika tetangga yang digunakan semakin banyak, model KNN akan lebih akurat namun waktu pengerjaannya juga akan lebih lama. Model KNN juga menggunakan *uniform weight* karena tidak ada titik atau baris yang perlu diberi bobot khusus. Bobot khusus ini umumnya diberikan jika sebaran datanya cukup *imbalance*. Secara intuisi, model KNN cocok digunakan karena keseluruhan data berada pada rentang yang sama yaitu [0,1]. Model KNN yang digunakan menghasilkan akurasi sebesar 88.86% dan *confusion matrix* yang dihasilkan adalah seperti pada gambar dibawah.

```
KNN Confusion Matrix :
 [[77  0  0  3  2  5  3  4]
 [ 0 63  3  0  3  4  1  1]
 [ 0  2 81  1  0  1  0  1]
 [ 0  1  0 46  2  0  0  6]
 [ 0  0  0  0 30  0  0  2]
 [ 0  1  0  0  0 47  0  4]
 [ 0  0  0  0  1  0 45  0]
 [ 0  0  0  1  0  0  0 26]]
```

Gambar 15. *Confusion matrix* untuk model KNN

Dari ketiga model diatas, terlihat pada *confusion matrix* bahwa kesalahan banyak terjadi pada prediksi kelas bukan 1 yang diprediksi sebagai kelas 1. Sebagai contoh, pada model *random forest,* terdapat 1 citra kelas 4 yang diprediksi sebagai kelas 1, 1 citra kelas 5 yang diprediksi sebagai kelas 1, dan seterusnya. Kesalahan ini disebabkan karena citra dengan kelas 1 merupakan citra terbanyak, sehingga model akan lebih mudah untuk memprediksi citra kelas 1 dan lebih sering salah dalam memprediksi kelas lain menjadi kelas 1.

Kesimpulan

Berdasarkan percobaan dari beberapa model yang diimplementasikan pada dataset, diketahui bahwa model terbaik untuk klasifikasi dan membedakan setiap *neckband* pada leher sapi adalah dengan menggunakan model *random forest*. Dengan menggunakan *default parameter* dari pustaka sklearn, diketahui bahwa model *random forest* mampu memprediksi secara akurat 91.87% citra pada data uji.

Daftar Pustaka

Alfian AN. JURUSAN TEKNIK INFORMATIKA FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG 2016. :102.

Anggraeny FT, Rahmat B, Pratama SP. 2020. Deteksi Ikan Dengan Menggunakan Algoritma Histogram of Oriented Gradients. *Inform. Mulawarman J. Ilm. Ilmu Komput.* 15(2):114.doi:10.30872/jim.v15i2.4648.

Avelita, B. (2013). Klasifikasi K-Nearest Neighbor. Dipetik 06 2016, 22, dari www.academia.edu:https://www.academia.edu/9131959/A._Klasifikasi_KNearest_Neighbor

Breiman dan Leo. 2001. *Machine Learning.* Berkeley:University of California

Chen R, Hawes M, Mihaylova L, Xiao J, Liu W. Vehicle Logo Recognition by Spatial-SIFT Combined with Logistic Regression. :9.

Dalal N, Triggs B. 2005. Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. [internet] Vol. 1. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05);. San Diego, CA, USA. San Diego, CA, USA: IEEE. hlm. 886–893. [diunduh 2022 Jun 17]. Tersedia pada:

Devella S, Yohannes Y, Rahmawati FN. 2020. Implementasi Random Forest Untuk Klasifikasi Motif Songket Palembang Berdasarkan SIFT. *JATISI J. Tek. Inform. Dan Sist. Inf.* 7(2):310–320.doi:10.35957/jatisi.v7i2.289.

Hasibuan B. 2016. PERLINDUNGAN HUKUM TERHADAP PETERNAK SAPI PERAH DIKAITKAN DENGAN KEBERADAAN ASOSIASI PETERNAK SAPI PERAH DALAM UPAYA MENINGKATKAN KESEJAHTERAAN PETERNAK. *J. Wawasan Yuridika*. 34(1):114.doi:10.25072/jwy.v34i1.112.

Hosmer, D.W dan S. Lemeshow. 2000. *Applied Logistic Regression. 2nd Edition*. New York: *John Willey* and Sons

James Blakely, David H. Bade. Diterjemahkan oleh Bambang Srigandono; Penyunting Soedarsono. *Ilmu Peternakan / BLAKELY, James* .1998

Nirwana. S.R.A. 2015. Regresi Logistik Multinomial dan Penerapannya dalam Menentukan Faktor yang Berpengaruh pada Pemilihan Program Studi di Jurusan Matematika UNM. Skripsi. Universitas Negeri Makassar. Makassar.

Ndaumanu, R. I. 2014. Analisis Prediksi Tingkat Pengunduran Diri Mahasiswa dengan Metode K-Nearest Neighbor. Jatisi.Vol 1, 3.

N. Dalal dan B. Triggs. 2005. *Histograms of Oriented Gradients for Human Detection* dalam *one-Alps*. Avenue de l'Europe. France:Montbonnot.

Santoso, B., 2007. *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis. 1st ed*. Yogyakarta: Graha Ilmu.

Sudono, A. 1999. *Ilmu Produksi Ternak Perah*. Bogor: Fakultas Peternakan Institut Pertanian Bogor.

Thepade SD, Bang SV, Chaudhari PR, Dindorkar MR. 2020. Covid19 Identification from Chest X-ray Images using Machine Learning Classifiers with GLCM Features. *ELCVIA Electron. Lett. Comput. Vis. Image Anal.* 19(3):85.doi:10.5565/rev/elcvia.1277.