

Deteksi *Defect Coffee* Pada Citra Tunggal *Green Beans* Menggunakan Metode *Ensamble Decision Tree*

Detection of Defect Coffee in Green Beans Single Image Using the Ensemble Decision Tree Method

Ami Rahmawati¹, Yan Rianto², Dwiza Riana³

^{1,2,3}Program Studi Ilmu Komputer, Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri

E-mail: ¹ami.amv@nusamandiri.ac.id, ²y_rianto@yahoo.co.jp, ³dwiza@nusamandiri.ac.id

Abstrak

Kopi merupakan salah satu komoditas minuman unggulan, sehingga permintaan biji kopi meningkat dari tahun ke tahun. Permintaan biji kopi didasarkan pada kualitas. Terdapat beberapa faktor yang mempengaruhi kualitas antara lain bagaimana kopi ditanam dan dipanen, adapun kurangnya nutrisi dan perlindungan tanaman yang tidak memadai, maka akan menghasilkan kopi yang berkualitas rendah. Biji kopi berkualitas rendah sering kali disebut *defects*. Identifikasi *defects coffee* sangat penting khususnya bagi para petani dan pengusaha kopi agar dapat memilih biji kopi yang berkualitas tinggi sehingga meningkatkan nilai jual biji kopi. Pada beberapa industri kopi maupun makanan, teknik untuk mengidentifikasi cacat biji kopi biasa dengan cara seleksi manual dan mekanik, yang mana membutuhkan waktu yang lama dan dapat merusak biji kopi. Oleh karena itu diperlukan suatu pendekatan yang lebih modern dalam mengidentifikasi cacat biji kopi seperti pengolahan citra. Untuk itu penelitian ini bertujuan melakukan pengolahan citra berupa segmentasi pada citra *green beans coffee* menggunakan metode *thresholding*. Setelah itu dilakukan analisis tekstur menggunakan GLCM (*Grey Level Co-occurrence Matrix*) dan dilanjutkan dengan pemodelan klasifikasi menggunakan algoritma C4.5 dengan *bagging*. Dari hasil penelitian yang diperoleh, akurasi dari penggunaan algoritma C4.5 dengan *bagging* sebesar 94%.

Kata kunci: *Defect Coffee*, *Thresholding*, Algoritma C4.5.

Abstract

Coffee is one of the leading beverage commodities, so the demand for coffee beans increases from year to year. Demand for coffee beans is based on quality. There are several factors that affect quality, including how coffee is grown and harvested, as for the lack of nutrition and inadequate plant protection, it will produce low quality coffee. Low quality coffee beans are often called defects. The identification of coffee defects is very important especially for farmers and coffee entrepreneurs to be able to choose high-quality coffee beans that increase the selling value of coffee beans. In some coffee and food industries, techniques for identifying defects of ordinary coffee beans by means of manual and mechanical selection, which takes a long time and can damage coffee beans. Therefore a more modern approach is needed in identifying coffee bean defects such as image processing. For this reason, this study aims at image processing in the form of segmentation of green beans coffee using the thresholding method. After that the texture analysis is done using GLCM (Gray Level Co-occurrence Matrix) and proceed with classification modeling using C4.5 algorithm with bagging. From the research results obtained, the accuracy of the use of the C4.5 algorithm with bagging is 94%.

Keywords: *Defect Coffee*, *Thresholding*, C4.5 Algorithm.

1. PENDAHULUAN

Kopi merupakan salah satu komoditas minuman unggulan yang diperdagangkan di pasar dunia maupun Indonesia, sehingga permintaan biji kopi meningkat dari tahun ke tahun [1]. Berdasarkan data dinas perkebunan pada tahun 2018 jumlah produksi kopi di Indonesia mencapai 722,461 ton dengan provinsi Jawa Barat sebagai salah satu dari sepuluh besar provinsi yang berkontribusi terhadap produksi kopi sebesar 19,602 ton [2].

Permintaan biji kopi didasarkan pada kualitas [3], yang merupakan aspek utama industri kopi untuk menghasilkan produk berkualitas tinggi yang dijadikan sebagai dasar kesuksesan pasar kompetitif saat ini [4]. Beberapa faktor yang mempengaruhi kualitas antara lain bagaimana kopi ditanam dan dipanen, adapun kurangnya nutrisi dan perlindungan tanaman yang tidak memadai, maka akan menghasilkan kopi yang berkualitas rendah [5]. Karakteristik sensasi rasa kopi yang tidak menyenangkan akibat biji kopi berkualitas rendah sering kali disebut *defects coffee* atau cacat kopi yang mana cacat tersebut dapat diketahui berdasarkan fisik, rasa dan aroma pada biji kopi [6]. Dalam *defects coffee* terdapat 14 jenis cacat diantaranya *insect damage*, *black*, *sour*, *faded*, dan *broken* [7]. Pada penelitian ini ada dua jenis *defects coffee* yang akan diidentifikasi yaitu *black* dan *insect damage*. Kedua jenis *defects coffee* tersebut dipilih karena memiliki total cacat lebih dari 4% sehingga tidak dapat ditoleransi kecacatannya [8].

Identifikasi *defects coffee* sangat penting khususnya bagi para petani dan pengusaha kopi agar dapat memilih biji kopi yang berkualitas tinggi sehingga meningkatkan nilai jual biji kopi [9]. Pada beberapa industri kopi maupun makanan, Teknik untuk mengidentifikasi cacat biji kopi biasa dengan cara seleksi manual dan mekanik. Teknik seleksi manual dilakukan oleh seorang ahli dengan menginspeksi visual terhadap biji kopi dan mengidentifikasi fitur spesifik dari biji kopi untuk memilih yang terbaik, namun pemilihan tidak seragam dikarenakan membutuhkan jam kerja yang lama [10]. Teknik mekanik dilakukan dengan mesin yakni mensortir biji kopi berdasarkan ukurannya, namun didalam teknik ini dapat berpotensi merusak biji kopi [11]. Oleh karena itu tidak ada teknik yang memberikan pilihan terbaik untuk memilih biji kopi berkualitas, yang berarti diperlukan suatu pendekatan yang lebih modern seperti pengolahan citra.

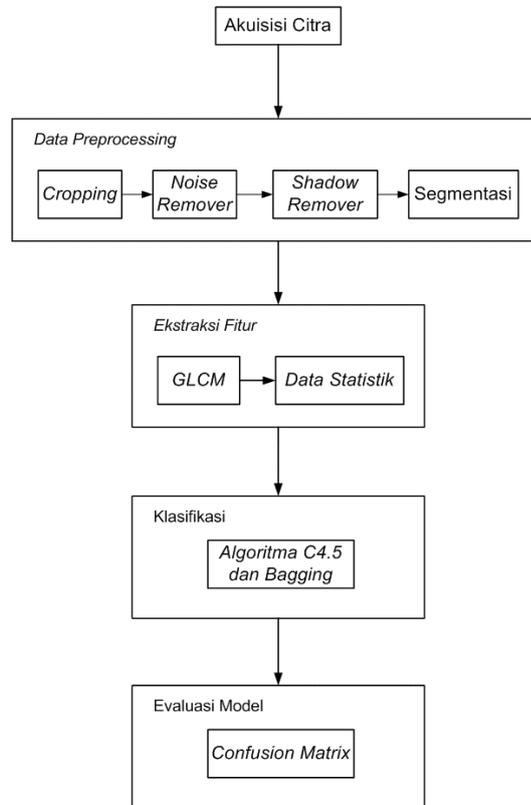
Ada banyak peneliti yang sudah melakukan penelitian untuk mengidentifikasi *defect coffee*, seperti penelitian yang dilakukan oleh [1] tentang algoritma KNN yang digunakan untuk menentukan kualitas biji kopi dan jenis cacat yang sesuai. Sistem ini terdiri dari logical proses, *image processing* dan algoritma *supervised learning* dengan menggunakan MATLAB. Penelitian selanjutnya dilakukan oleh [12] tentang identifikasi kualitas biji kopi menggunakan teknik *image processing* melalui ekstraksi komponen warna RGB dari sebuah citra. Penelitian selanjutnya dilakukan oleh [9] tentang perancangan aplikasi yang dapat mendeteksi kualitas biji kopi mentah baik dan buruk berdasarkan nilai HSI (*Hue Saturation Intensity*) pada biji kopi menggunakan pengolahan citra digital dan jaringan syaraf tiruan *Backpropagation*. Penelitian selanjutnya dilakukan oleh [7] tentang pemeriksaan cacat biji dengan *GAN-Structured Automated Labeled Data Augmentation* di industri kopi menggunakan *deep learning*, *Generative-Adversarial Network*, *Automated Labeled Data Augmentation*. Penelitian selanjutnya dilakukan oleh [13] tentang Meningkatkan Kualitas Pemeriksaan Cacat Jaringan dengan *Deep-Learning* pada Kacang Padat dengan Menggunakan *Hough Circle Transform* untuk Industri Kopi. Penelitian selanjutnya dilakukan oleh [14] tentang Klasifikasi Citra Biji Kopi Hijau Berdasarkan Jenis Cacat Menggunakan *Convolutional Neural Network* (CNN).

Merujuk pada hasil penelitian-penelitian tersebut, maka pada penelitian ini akan mendeteksi *defect coffee* pada citra tunggal yang didapatkan dari bumi kopi goalpara sukabumi dengan menggunakan *machine learning*. *Machine learning* pada saat ini menjadi jalan yang optimal dengan tujuan untuk mendapatkan algoritma terlatih yang dapat mengidentifikasi cacat biji kopi [15]. Untuk dapat mengidentifikasi cacat biji kopi tersebut diperlukan kualitas citra yang baik. Oleh karena itu pada penelitian ini akan dilakukan tahapan *preprocessing* meliputi *noise remover*, *shadow remover*, dan segmentasi yang bertujuan untuk menghasilkan kualitas citra yang diharapkan. Selain itu, teknik *machine learning* yang akan digunakan yaitu klasifikasi

Algoritma *Decision Tree* (C4.5) dengan *Bagging* yang nantinya akan menunjukkan keakuratan dari hasil prediksi cacat biji kopi.

2. METODE PENELITIAN

2.1 Tahapan Penelitian



Gambar 1. Model Tahapan Penelitian

Pada tahapan penelitian diatas menjelaskan bahwa untuk mendeteksi *defect coffee* pada citra *green beans* dimulai dari akuisisi citra kemudian dilakukan *data preprocessing* meliputi *cropping*, *noise remover* menggunakan *Gaussian*, *shadow remover* menggunakan *Absolute Difference* dan segmentasi menggunakan metode *Thresholding* terhadap citra tersebut. Selanjutnya citra yang sudah tersegmentasi diekstraksi menggunakan *GLCM* (*Gray Level Co-occurrence Matrix*) untuk mendapatkan fitur dan data statistik. Setelah itu data akan diuji dengan metode data mining klasifikasi algoritma C4.5 dan teknik *Bagging* yang akan divalidasi dengan menggunakan *confusion matrix* untuk mengetahui nilai akurasi.

2.2 Gaussian

Filter gaussian adalah alat perataan yang efisien dalam visi komputer. Berdasarkan pada kernel gaussian, Witkin [Witkin (1983)] dan Koenderink secara resmi mendefinisikan konsep ruang-skala pada tahun 1983 - 1984, menyatukan serangkaian gambar turunan yang tidak terkait pada berbagai tingkat detail sepanjang parameter skala dan memungkinkan data gambar untuk menjadi diintegrasikan ke dalam perspektif yang berkelanjutan. Sejak itu, kernel Gaussian telah sering dipelajari dalam hubungannya dengan apa yang disebut "*multiple scale approach*". Namun *scale-space filtering* telah diterapkan dalam gambar 2D [Kuijper dan Florack (2001)] [Bauer dan Peikert (2002)] [Lindberg (1995)] [Linberg (1998)], dan baru-baru ini saja teknik ini digunakan dalam grafik komputer, dengan aplikasi terbatas dalam visualisasi 3D. Filter intrinsik, generalisasi *scale-space filtering*, digunakan untuk menghilangkan noise dari gambar dan data yang dipindai [Boulanger et al. (2002)]. Peneliti yang relatif lebih sedikit telah melihat

ke penyederhanaan model 3D (atau mesh) berdasarkan analisis titik fitur pada berbagai skala [16].

Dalam satu dimensi kernel $G\sigma(x)$ dari filter gaussian domain diberikan oleh fungsi gaussian [17]

$$G\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/(2\sigma^2)} \quad (1)$$

Di mana σ adalah sebuah konstanta, yang disebut *variance function* dalam dua dimensi, kernel didefinisikan oleh [17]

$$G\sigma(x, y) = \frac{1}{2\sigma^2\pi} e^{-(x^2+y^2)/(2\sigma^2)} \quad (2)$$

Analisis cepat dalam domain kontinu menunjukkan bahwa filter gaussian adalah filter lowpass. Transformasi fourier dari distribusi gaussian juga merupakan gaussian. fungsi transfer bersifat gaussian, sehingga frekuensi tinggi dalam sinyal yang disaring diredam oleh faktor yang tumbuh secara eksponensial dengan frekuensi [17].

2.3 Absolute Difference

Ukuran perbedaan antara gambar berfungsi sebagai fitur untuk mengukur kesamaan, terdapat dua metode dasar untuk perbedaan gambar: perbedaan *absoulte* dan perbedaan histogram. *Absolute difference* membutuhkan lebih sedikit perhitungan, tetapi umumnya lebih rentan terhadap noise dan artefak pencitraan lainnya. Perbedaan gambar dari dua gambar didefinisikan sebagai jumlah dari perbedaan *absolute* pada setiap piksel. Gambar pertama dianalisis dengan gambar kedua, I_{t-T} , pada jarak temporal T. Nilai *difference* didefinisikan sebagai berikut [18] :

$$D(t = \sum_{i=1}^M |I_{t-T}(i) - I_t(i)|) \quad (3)$$

Dimana M adalah resolusi, atau jumlah piksel dalam gambar. Metode ini untuk perbedaan gambar *noise* dan sangat sensitif terhadap gerakan kamera dan degradasi gambar [18].

2.4 GLCM (Gray Level Co-occurrence Matrix)

GLCM atau matriks dependensi spasial *gray-tone*. Konsep utama GLCM adalah bahwa informasi tekstur yang terkandung dalam gambar ditentukan oleh hubungan kedekatan yang harus dimiliki *gray-tone* dalam gambar. Dengan kata lain, diasumsikan bahwa informasi tekstur ditentukan oleh nilai f_{ij} dalam GLCM, di mana f_{ij} menunjukkan frekuensi kemunculan dua sel tone abu-abu i dan j, masing-masing, dipisahkan oleh jarak d dengan arah khusus pada gambar. Nilai f_{ij} dapat dihitung untuk setiap arah dan jarak yang layak d. Umumnya, hanya empat arah yang sesuai dengan sudut $0^\circ, 45^\circ, 90^\circ, 135^\circ$ yang digunakan [19].

2.5 Algoritma C4.5

Algoritma ini diperkenalkan oleh Quinlan (1996) untuk mengatasi kelemahan dari ID3 yang tidak dapat menangani atribut bertipe numerik. Pada prinsipnya, cara kerja algoritma C4.5 sama dengan ID3, yaitu menggunakan nilai *Gain* dalam menentukan fitur/atribut yang menjadi pemecah node pada pohon keputusan. Perbedaannya adalah algoritma C4.5 dapat menangani atribut bertipe numerik dengan cara melakukan diskretisasi data terhadap atribut bertipe numerik. Pendekatan untuk menyatakan syarat pengujian pada node terdiri atas [20] :

1. Pemecahan pada Atribut Binomial/Biner
2. Pemecahan pada Atribut Nominal (Kategorikal)
3. Pemecahan pada Atribut Numerik

Proses pembentukan pohon keputusan dengan algoritma C4.5 yaitu [21] :

1. *Entropy(S)* adalah parameter yang digunakan untuk mengukur keberagaman setiap nilai *attribute* kriteria terhadap *decision attribute* (atribut keputusan) dalam sebuah kumpulan data. Makin rendah nilai *Entropy* maka tingkat keberagaman suatu kumpulan data makin rendah, sebaliknya makin tinggi nilai *Entropy* maka nilai keberagaman makin tinggi. Rumus *Entropy* sebagai berikut:

$$Entropy(S) = \sum_{i=1}^n -p_i * \log_2(p_i) \quad (4)$$

Keterangan :

S = Jumlah sampel data (*Sampling*)

n = Jumlah partisi S

p_i = Proporsi dari S_i terhadap S

2. *Gain (S,A)* adalah selisih nilai *Entropy* total dikurangi nilai *Entropy* masing-masing nilai setiap *attribute* kriteria dikali nilai proporsi nilai *attribute* dibagi jumlah sampel data. Fungsi dari nilai *Gain* adalah untuk mengukur efektivitas masing-masing *attribute* kriteria dalam mengklasifikasikan data. Pada algoritma C4.5 nilai *Gain* digunakan sebagai dasar pembentukan *node* atau akar dan cabang pohon keputusan. Rumus *Gain* adalah sebagai berikut:

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \quad (5)$$

Keterangan :

S = Jumlah kasus (*Sampling*)

A = *Attribute*

n = Jumlah partisi S

$|S_i|$ = Jumlah kasus pada partisi ke- i

$|S|$ = Jumlah kasus dalam S

3. HASIL DAN PEMBAHASAN

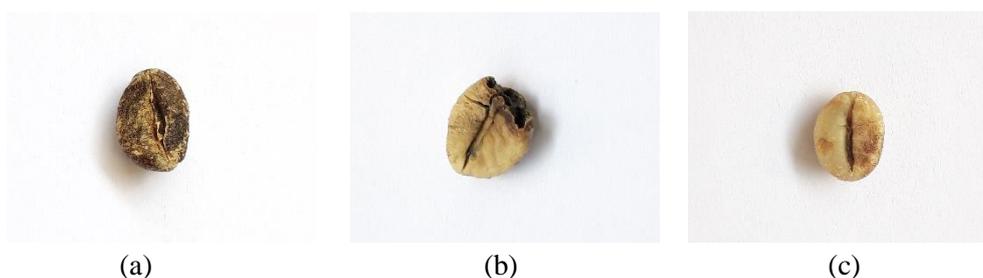
Penelitian ini bertujuan untuk memperoleh hasil akurasi terbaik dengan menggunakan model algoritma C4.5 dan bagging dalam mendeteksi *defect coffee* berdasarkan kumpulan-kumpulan *image* yang telah di ekstraksi.

a. Dataset

Data yang digunakan pada penelitian ini menggunakan data primer. Data penelitian diambil dengan cara memotret *defect* pada *green beans coffee* yang diperoleh dari Bumi Kopi Goalpara Sukabumi, dengan jumlah 400 *image* yang terdiri dari 150 *image* untuk *black coffee*, 150 *image* untuk *damage coffee*, dan 100 *image* untuk normal (*non-defect coffee*).

b. Akuisisi Citra

Pengambilan gambar *defect coffee* dilakukan menggunakan kamera *smartphone* Samsung galaxy A6 dengan resolusi 16 MP dengan piksel 4608 x 3456. Gambar *defect coffee* diambil secara tunggal dan ditempatkan pada latar belakang putih. Kamera diatur pada zoom 4,0x ditahan dalam posisi normal pada bidang biji *defect coffee* dengan jarak 5,1 inchi tepat diatas bidang sampel. Gambar tersebut disimpan dalam format JPG. Citra *defect coffee* tersebut diambil dari bumi kopi goalpara sukabumi yang ditunjukkan pada gambar 2 berikut:



Gambar 2. Sampel Citra *Defect Coffee*

a. *Black Coffee*; b. *Damaged Coffee*; c. *Non-Defect Coffee*

c. *Cropping*

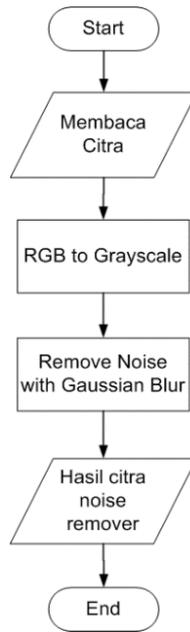
Pada tahap *preprocessing* dilakukan proses *cropping* pada citra dikarenakan gambar yang dihasilkan oleh kamera cukup besar. Ukuran piksel 4608 x 3456 dirubah menjadi 2500 x 2400. Perubahan ukuran tersebut ditunjukkan pada gambar 3 berikut:



Gambar 3. *Image Cropping*

d. *Noise Remover*

Gambar-gambar yang telah diambil oleh kamera terdapat *noise*. *Noise* tersebut dapat menyebabkan pemrosesan gambar menjadi buruk. Untuk menghilangkan *noise* yang terdapat pada gambar dalam penelitian ini menggunakan *Gaussian Blur* dalam *opencv*. *Gaussian Blur* akan memindai setiap piksel gambar dan menghitung ulang nilai piksel berdasarkan nilai piksel yang mengelilinginya, area yang di pindai di sekitar setiap piksel disebut dengan *kernel*. Langkah-langkah menghilangkan *noise* dengan *Gaussian Blur* disajikan dalam *flowchart* (Gambar 4) sebagai berikut :



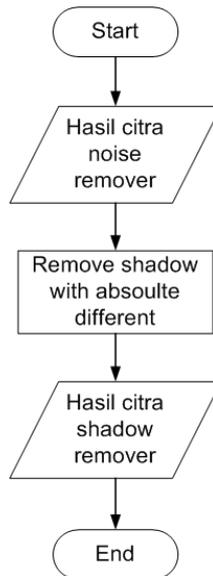
Gambar 4. Flowchart Noise Remover

Penjelasan untuk *flowchart noise remover* pada gambar 4.3 adalah :

1. Membaca citra.
2. Citra asal yang bertipe RGB dikonversi menjadi *grayscale*. Citra *grayscale* merupakan citra yang nilai intensitas pikselnya berdasarkan derajat keabuan. Konversi citra dilakukan untuk memudahkan proses pada metode *Gaussian Blur*.
3. *Gaussian Blur* memfilter gambar 2D yaitu citra *grayscale* dengan ukuran kernel 51 x 51 dan *standard deviation* adalah 0.
4. Menampilkan hasil citra.

e. *Shadow Remover*

Untuk menghilangkan bayangan pada gambar dalam penelitian ini menggunakan *Absolute Difference* dalam *opencv*. *Absolute Difference* digunakan untuk mengurangi informasi yang tidak perlu termasuk dapat menghilangkan bayangan dalam sebuah gambar dengan cara menemukan perbedaan absolut antara piksel dari dua *array* gambar. Langkah-langkah menghilangkan bayangan dengan *Absolute Difference* disajikan dalam *flowchart* (Gambar 5) sebagai berikut :



Gambar 5. Flowchart Shadow Remover

Penjelasan untuk *flowchart shadow remover* pada gambar 4.4 adalah :

1. Membaca citra dari hasil *noise remover*.
2. *Absolute Difference* digunakan dengan cara mengurangi nilai 255 dengan citra *grayscale* dan citra *noise remover*.
3. Menampilkan hasil citra.

3.6 Segmentasi

Segmentasi citra yang digunakan adalah teknik *Thresholding*. Segmentasi citra mengacu pada proses membagi gambar menjadi beberapa bagian untuk tujuan membedakan objek dengan *background* yang mana diperlukan untuk menjalankan ekstraksi fitur. Proses segmentasi pada penelitian ini menghapus *background* dan mengganti *background* tersebut menjadi hitam akan tetapi untuk warna objek tetap yaitu RGB, maka digunakan metode deteksi warna. Setelah semua ruang warna di coba maka HSV (*Hue*, *Saturation*, dan *Value*) dan *grayscale* menjadi pilihan terbaik untuk proses segmentasi *defect coffee beans*. Hasil proses segmentasi ditunjukkan pada gambar 6 berikut :



Gambar 6. Hasil Segmentasi

3.7 Ekstraksi Fitur

Untuk ekstraksi fitur yang digunakan adalah GLCM (*Gray Level Co-occurrence Matrix*). GLCM akan menghasilkan matriks baru dengan cara menghitung nilai kemungkinan pada piksel tetangganya. Matriks yang terbentuk berjumlah 4 dengan sudut 0, 45, 90, dan 135 derajat, akan tetapi yang digunakan pada penelitian ini sudut 0°. Proses ekstraksi tersebut menghasilkan 6 (enam) fitur yaitu *Contrast*, *Correlation*, *Homogeneity*, *ASM*, *Disimilarity*, *Energy*. Setelah fitur diketahui maka selanjutnya menghitung rata-rata data statistik. Hasil

perhitungan tersebut disimpan dalam format CSV dan akan menjadi data input untuk data latih dan data uji yang ditunjukkan pada tabel 1 berikut :

Tabel 1. Hasil Ekstraksi Fitur

Contrast	Correlation	Homogeneity	ASM	Disimilarity	Energy	Class
0.294891	0.974451	0.934721	0.520494	0.149324	0.72145	Black Coffee
0.334647	0.973339	0.927062	0.486048	0.167601	0.697168	Black Coffee
0.322861	0.9739	0.928864	0.487774	0.162938	0.698405	Black Coffee
0.314337	0.972303	0.926171	0.502878	0.167707	0.709136	Black Coffee
0.313828	0.971834	0.92691	0.481521	0.165699	0.693913	Black Coffee
0.139104	0.992309	0.97422	0.638414	0.060322	0.799008	Damaged Coffee
0.185369	0.987164	0.965894	0.569974	0.079472	0.754965	Damaged Coffee
0.235411	0.968954	0.958226	0.628033	0.099769	0.792484	Damaged Coffee
0.275008	0.966241	0.956264	0.619025	0.107167	0.78678	Damaged Coffee
0.217068	0.975394	0.957868	0.604277	0.098135	0.777351	Damaged Coffee
0.140321	0.993177	0.977415	0.63876	0.054647	0.799224	Normal Coffee
0.137985	0.993353	0.977533	0.63835	0.054119	0.798968	Normal Coffee
0.150022	0.992657	0.977172	0.637856	0.055891	0.798659	Normal Coffee
0.212686	0.991013	0.967396	0.5094	0.078998	0.713721	Normal Coffee
0.184119	0.990808	0.975243	0.583269	0.062008	0.76372	Normal Coffee

3.8 Hasil pengujian Model C4.5 dengan Bagging

Hasil dari uji coba yang dilakukan yaitu untuk menghasilkan nilai *accuracy* (*confusion matrix*). Model *confusion matrix* akan membentuk matrix yang terdiri dari *true positif* atau tupel positif dan *true negatif* atau tupel negatif, kemudian masukan data *testing* yang sudah disiapkan ke dalam *confusion matrix* sehingga didapatkan hasil pada tabel 2 di bawah ini:

Tabel 2. Confusion Matrix Algoritma C4.5 dengan Bagging

	Black	Damaged	Normal
Black	48	0	0
Damaged	1	38	2
Normal	0	3	28

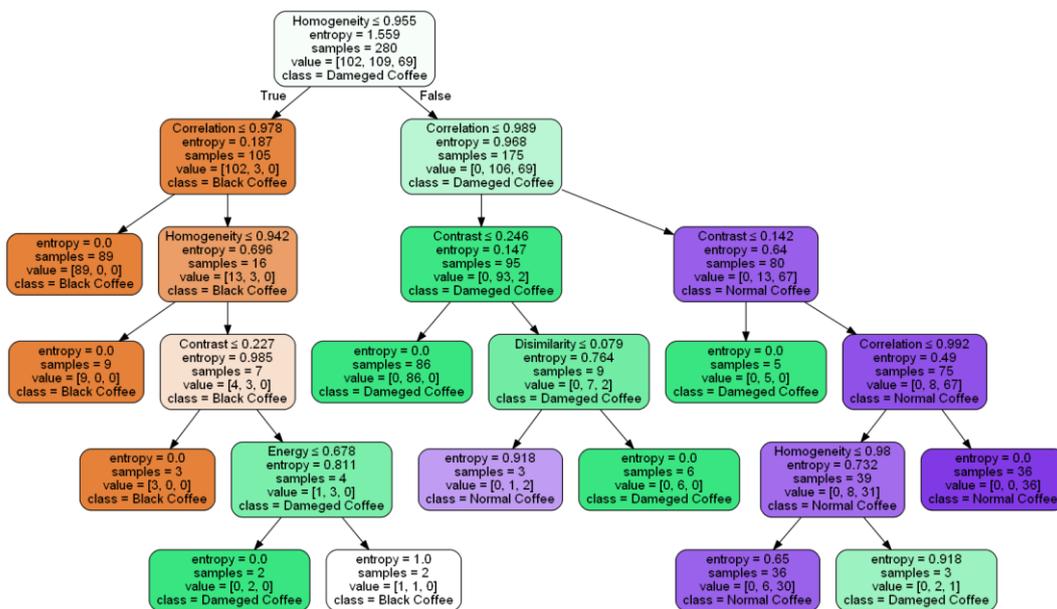
Berdasarkan Tabel 2 dari 400 tupel dibagi menjadi data latih dan data uji dengan perbandingan pembagian yaitu 70:30 maka data yang digunakan untuk prediksi uji sebanyak 30% dari jumlah data masing-masing *class*. Terdapat rincian jumlah *True Positive (TP) Black* 48, *False Negative (FN) Black* 1, *False Positive (FP) Black* 0, dan *True Negative (TN) Black* 71. *True Positive (TP) Damaged* 38, *False Negative (FN) Damaged* 3, *False Positive (FP) Damaged* 3, dan *True Negative (TN) Damaged* 76. *True Positive (TP) Normal* 28, *False Negative (FN) Normal* 2, *False Positive (FP) Normal* 3, dan *True Negative (TN) Normal* 87. Dari data tersebut maka dapat dihitung nilai *accuracy*, *precision*, *recall*, *f1-score*, dan *support*. Data hasil olahan dapat dilihat pada gambar 7 berikut :

Accuracy C4.5 with Bagging: 0.9416666666666667
 Classification C4.5 with Bagging:

	precision	recall	f1-score	support
Black Coffee	0.98	1.00	0.99	48
Damaged Coffee	0.93	0.93	0.93	41
Normal Coffee	0.93	0.90	0.92	31
accuracy			0.95	120
macro avg	0.95	0.94	0.94	120
weighted avg	0.95	0.95	0.95	120

Gambar 7. Nilai Accuracy, Precision, Recall, F1-score, dan Support

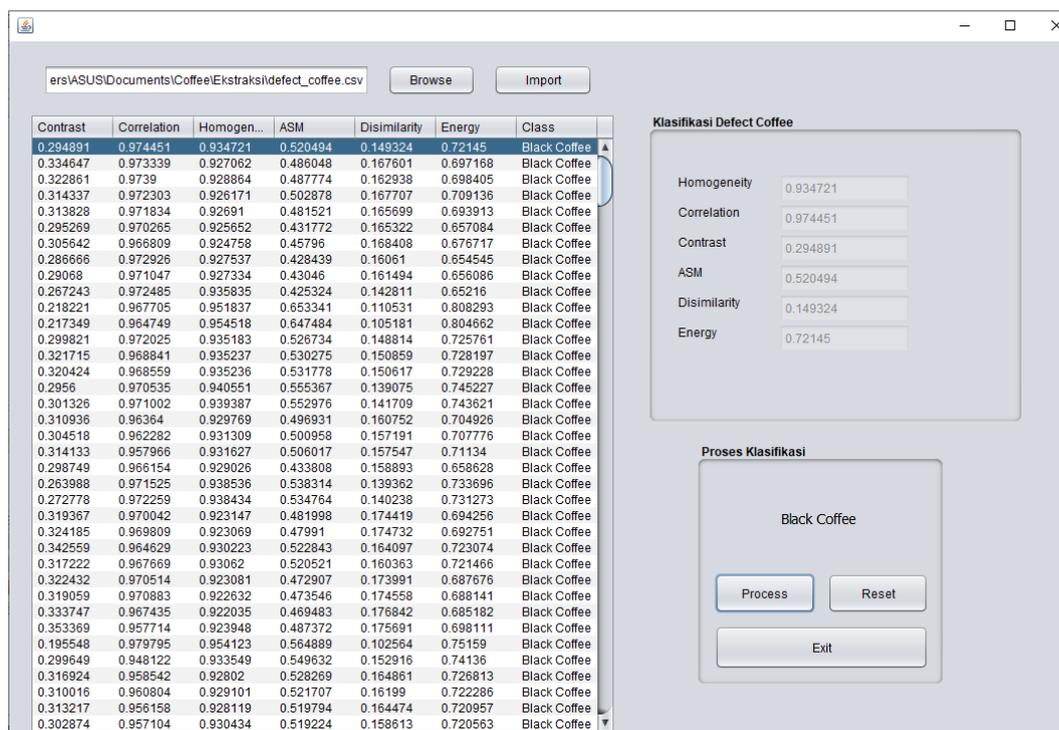
Berdasarkan gambar 7 bahwa pengujian menggunakan algoritma C4.5 dengan *bagging* menghasilkan akurasi 94% dan pohon keputusan yang terbentuk (lihat Gambar 8), dimana pohon keputusan tersebut terbentuk dengan memanfaatkan Framework Python 3.7.1.



Gambar 8. Pohon Keputusan Klasifikasi Defect Coffee Beans

3.9 Pengembangan Aplikasi Model C4.5 dengan Bangging

Model yang dihasilkan dari eksperimen dikembangkan menjadi sebuah aplikasi. Aplikasi tersebut akan dikembangkan menggunakan bahasa pemrograman JAVA berbasis desktop yang akan dibuat dengan Netbeans 7.4.



Gambar 9. Pengujian Aplikasi Model Algoritma C4.5 dengan Bagging

4. KESIMPULAN DAN SARAN

Dari hasil penelitian yang diperoleh, metode segmentasi yang cocok untuk mendeteksi *defect coffee beans* yaitu *Thresholding* dengan deteksi warna citra HSV (*Hue*, *Saturation*, dan *Value*) dan *grayscale*. Untuk ekstraksi fitur yang digunakan yaitu GLCM (*Gray Level Co-occurrence Matrix*). Akurasi dari penggunaan algoritma klasifikasi C4.5 dengan *bagging* dalam deteksi *defect coffee beans* sebesar 94%. Berdasarkan penjelasan tersebut, penerapan metode *Thresholding* mampu mensegmentasi citra dengan baik, sehingga pada saat diuji menggunakan algoritma C4.5 dengan *bagging* menghasilkan tingkat akurasi yang baik.

DAFTAR PUSTAKA

- [1] M. Garc and J. E. Candelo-becerra, "Quality and Defect Inspection of Green Coffee Beans Using a Computer Vision System," *MDPI*, vol. 9, pp. 2–18, 2019.
- [2] Direktorat Jenderal Perkebunan, "Produksi Kopi Menurut Provinsi di Indonesia , 2015-2019 Coffee," vol. 2019, p. 2019, 2019.
- [3] N. Caporaso, M. B. Whitworth, S. Grebby, and I. D. Fisk, "Rapid prediction of single green coffee bean moisture and lipid content by hyperspectral imaging," *J. Food Eng.*, vol. 227, pp. 18–29, 2018.
- [4] D. Fernandes, A. Lucia, D. S. Madureira, D. Sun, S. Lucy, and E. Yoko, "Application of infrared spectral techniques on quality and compositional attributes of coffee : An overview," *FRIN*, vol. 61, pp. 23–32, 2014.
- [5] E. R. Arboleda, "An Image Processing Technique for Coffee Black Beans Identification," *2018 IEEE Int. Conf. Innov. Res. Dev.*, pp. 1–5, 2018.
- [6] Coffeeland Indonesia, "MENGIDENTIFIKASI DEFECT PADA KOPI," *Coffeeland*, 2020. [Online]. Available: coffeeland.co.id/.
- [7] Y. Chou *et al.*, "Deep-Learning-Based Defective Bean Inspection with GAN-Structured Automated Labeled Data Augmentation in Coffee Industry," *MDPI*, vol. 9, pp. 2–26,

- 2019.
- [8] Dani, "Seputar Kopi," 2020.
 - [9] H. A. L. Nasution H T, Rumansa H, "Designing the quality of coffee bean detection application using Hue Saturation Intensity Designing the quality of coffee bean detection application using Hue Saturation Intensity," *IOP Conf. Ser. Mater. Sci. Eng.*, no. 648, pp. 1–7, 2019.
 - [10] N. Razmjoooy, B. S. Mousavi, and F. Soleymani, "A real-time mathematical computer method for potato inspection using machine vision," *Comput. Math. with Appl.*, vol. 63, no. 1, pp. 268–279, 2012.
 - [11] M. Baigvand, A. Banakar, S. Minaei, J. Khodaei, and N. Behrooz-khazaei, "Machine vision system for grading of dried figs," *Comput. Electron. Agric.*, vol. 119, pp. 158–165, 2015.
 - [12] E. R. Arboleda, A. C. Fajardo, and R. P. Medina, "Classification of Coffee Bean Species Using Image Processing , Artificial Neural Network and K Nearest Neighbors," *2018 IEEE Int. Conf. Innov. Res. Dev.*, vol. 18, pp. 1–5, 2018.
 - [13] C. K. D. Wang *et al.*, "Improving Defect Inspection Quality of Deep-Learning Network in Dense Beans by Using Hough Circle Transform for Coffee Industry," *2019 IEEE Int. Conf. Syst. Man Cybern.*, vol. 19, pp. 798–805, 2019.
 - [14] C. Pinto, J. Furukawa, H. Fukai, and S. Tamura, "Classification of Green Coffee Bean Images Based on Defect Types Using Convolutional Neural Network (CNN)," *2017 Int. Conf. Adv. Informatics, Concepts, Theory, Appl.*, vol. 17, pp. 1–5, 2017.
 - [15] M. D. Podolsky, A. A. Barchuk, V. I. Kuznetcov, F. Natalia, V. S. Gaidukov, and S. A. Tarakanov, "Evaluation of Machine Learning Algorithm Utilization for Lung Cancer Classification Based on Gene Expression Levels," vol. 17, pp. 835–838, 2016.
 - [16] I. Cheng, G. M. Cortelazzo, A. Basu, and S. K. Tripathi, *3D Online Multimedia & Games: Processing, Transmission and Visualization*. Singapura: World Scientific, 2008.
 - [17] L. Velho, A. C. Frery, and J. Gomes, *Image Processing for Computer Graphics and Vision*. London: Springer, 2009.
 - [18] A. C. Bovik, *Handbook of Image and Video Processing*. London: Elsevier, 2010.
 - [19] B. Tso and P. M. Mather, *Classification Methods for Remotely Sensed Data*. London: Taylor & Francis, 2001.
 - [20] B. Santoso, A. I. S. Azis, and Zohrahayaty, *Machine Learning & Reasoning Fuzzy Logic Algoritma, Manual, Matlab, & Rapid Miner*. Yogyakarta: Deepublish, 2020.
 - [21] E. Buulolo, *Data Mining Untuk Perguruan Tinggi*. Yogyakarta: Deepublish, 2020.