MENGHITUNG OBYEK 2D MENGGUNAKAN CONNECTED COMPONENT LABELING

Kukuh Yudhistiro
Fakultas Teknologi Informasi Universitas Merdeka Malang
kukuh.yudhistiro@unmer.ac.id

Abstrak

Perhitungan obyek dalam bentuk 2 dimensi (2D) sering dilakukan pada sebuah citra foto. Contoh kasus yang membutuhkan deteksi dan perhitungan obyek pada sebuah citra foto adalah obyek sel tubuh seperti sel darah serta sel tubuh lainnya, kontur sebuah wilayah dan obyek biji-bijian. Tahap deteksi obyek pada sebuah citra harus terlebih dahulu dilakukan. Pada umumnya obyek dalam sebuah citra memiliki keseragaman warna piksel. Algoritma *connected component labeling* (CCL) adalah metode yang dapat digunakan untuk mengklasifikasikan *region* atau objek dalam citra digital. Algoritma ini menerapkan teori *connectivity* piksel dari citra. Seluruh piksel pada sebuah *region* disebut *connected* atau memiliki hubungan bila mematuhi aturan *adjacency* atau "kedekatan" piksel. Aturan kedekatan piksel ini memanfaatkan ketetanggaan antara piksel satu dengan piksel yang lainnya. Citra yang dapat diolah dengan menggunakan algoritma CCL adalah citra biner atau citra monokrom. Paper ini juga menggunakan metode *image preprocessing* lain seperti *binary thresholding* dan *median filter* sebelum dilakukan CCL agar hasil deteksi dan perhitungan dapat mencapai akurasi di atas 80%.

Kata kunci : *connected component labeling, adjacency, 2D object counting*

Abstract

The 2D object detection and counting are a challenging problem in image analysis. We need the object detection, recognition and counting in case of the body cells, blood cells, the contour of a region, grain objects or another object in an image. In this paper, object detection is the first stage that must first be done in pixel color uniformity image. In this paper, the connected component labeling (CCL), one of the most important techniques develop in order to count the object. This algorithm applies the connectivity pixel in an image. All pixels in a region are called connected or have a relationship when they have pixel adjacency. The rule of pixel adjacency use the neighbors between one pixel with another. The image that can be processed by using CCL algorithm is binary image or monocrhome. Another image processing techniques are applies such as binary thresholding and median filters before applying the CCL algorithm so that the results of detection and calculation can achieve accuracy above 80%.

***Keyword** : connected component labeling, adjacency, 2D object counting*

 ISSN : 2597 - 4696

PENDAHULUAN

Penelitian ini fokus pada deteksi fitur obyek 2 dimensi pada sebuah citra biji jagung untuk menghitung prosentase cemaran aflatoksin menggunakan CCL dan beberapa algoritma preprocessing seperti *binary filter, median filter* dan *image convolution*. Aflatoksin merupakan racun dari parasit pada jagung yang hanya terlihat dibawah sinar ultraviolet (UV). Aflatoksin dapat menyebabkan *nekrosis* akut, *cirrhosis* dan *carcinoma* pada hati hewan. Pada manusia diduga menimbulkan *carcino genic* (kanker hati). Tidak ada hewan yang resisten terhadap efek racun dari aflatoksin, demikian juga manusia. Oleh karena itu, melalui paper ini peneliti hendak mencoba salah satu metode untuk mengetahui prosentase cemaran tersebut dengan menghitung jumlah biji yang tercemar dan yang tidak tercemar. Untuk mendeteksi bijian yang tercemar, menggunakan foto UV sedangkan untuk menghitung keseluruhan biji jagung menggunakan foto normal. Perumusan masalah dapat diilustrasikan dengan contoh kasus pada seleksi jagung yang disimpan di gudang untuk siap dijual ke pabrik pengolahan. Pertama-tama pihak gudang harus mengetahui berapa prosentase cemaran aflatoksin terhadap jagung pipilan yang telah disimpan tersebut. Untuk mengetahui prosentase tersebut, maka harus dilakukan perhitungan jumlah jagung yang tercemar dan yang tidak tercemar. Bila dilakukan secara manual, maka tidak efisien dan efektif dalam penggunaan tenaga maupun waktu. Dengan mendeteksi fitur jagung untuk menghitung jagung pipilan dan aflatoksin secara otomatis ini, diharapkan mampu memberikan informasi yang mendekati akurat tentang prosentase cemaran aflatoksin dari jagung yang disimpan tersebut dengan cepat.

Tujuan dan manfaat dari penelitian yang akan dilakukan terhadap topik yang diambil ini adalah mengetahui prosentase cemaran aflatoksin pada jagung pipilan yang diperlukan oleh petani atau suplier sebagai alat ukur kualitas jagung sebelum dijual ke pabrik/industri.

KAJIAN LITERATUR

Algoritma *connected component labeling* adalah metode yang dapat digunakan untuk mengklasifikasikan *region* atau objek dalam citra digital. Algoritma ini menerapkan teori *connectivity* piksel dari citra. Seluruh piksel pada sebuah *region* disebut

ISSN : 2597 - 4696

connected atau memiliki hubungan bila mematuhi aturan *adjacency* atau "kedekatan" piksel. Aturan kedekatan piksel ini memanfaatkan ketetanggaan antara piksel satu dengan piksel yang lainnya. Oleh karena itu setiap piksel yang bersifat *connected* pada dasarnya memiliki *adjacency* satu sama lain karena mempunyai hubungan ketetangggan atau *neighbourhood*. Citra yang dapat diolah dengan menggunakan algoritma *connected component labeling* ini adalah citra biner atau citra monokrom. Selain itu, ketetanggaan harus memiliki panjang atau jarak 1 unit atau bersifat langsung antara piksel satu dengan yang lain tanpa ada perantaranya.

Menurut Gonzales dan Woods (1992, p40), terdapat dua macam konektivitas yang digunakan pada citra 2 dimensi yaitu:

- *4-Connected Neighbors*
- *8-Connected Neighbors*

Berikut algoritma CCL:

Terlebih dahulu siapkan array misal bernama CC (kependekan dari *Connected Component*) dan array bernama Temp. Array CC adalah *nested array* yang berfungsi menyimpan array Temp yang menyimpan indeks-indeks piksel yang saling bertetangga dan membentuk satu buah obyek. Jadi jumlah obyek yang dapat dideteksi pada sebuah citra diambil dari jumlah elemen array CC. Sedangkan kumpulan indeks yang merupakan jumlah piksel yang membentuk satu obyek disimpan pada array Temp. Sehingga dapat dikatakan satu array Temp mewakili satu obyek.

Algoritma 1 Tahap CCL

[Tahapan untuk deteksi obyek yang memiliki nilai kesamaan 1 atau 0]

Step 1:

- Cek piksel pada array biner secara berurutan
- Jika piksel bernilai *foreground* atau 1 maka ambil indeks piksel tersebut, lanjut ke Step 2.
- Cek piksel berikutnya

Step 2:

- Buat array baru, misal bernama Temp
- Tambahkan indeks piksel yang diperoleh pada langkah 1 sebagai elemen array Temp.
- Ubah nilai piksel dari 1 ke 0 pada array biner citra (agar tidak terbaca ulang)

ISSN : 2597 - 4696

- Cek semua 8 tetangga dari piksel yang tersimpan pada array Temp,
- jika nilai dari piksel tersebut 1 maka tambahkan indeks tetangga tersebut ke array Temp
- Ubah piksel tetangga dan inti ke 0 pada array biner citra
- Bila tidak ditemukan lagi piksel bernilai 1 dari 8 tetangga dari setiap indeks pada array Temp, maka tambahkan atau simpan array Temp ke array CC.
- Kembali ke Step 1 poin 3.

Berikut contoh penerapan algoritma di atas pada citra yang berisi karakter "Hi". Citra ini memiliki tiga buah obyek yakni "H", "|", dan ".". Dengan 10 citra 100, panjang array biner citra 0 hingga 99.

a. Pada proses *scanning*, jika piksel tersebut bernilai 1 maka simpan indeks piksel ke array Temp.

Temp=[11]

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	**1**	0	**1**	0	0	**1**	0	0	0
0	**1**	0	**1**	0	0	0	0	0	0
0	**1**	**1**	**1**	0	0	**1**	0	0	0
0	**1**	0	**1**	0	0	**1**	0	0	0
0	**1**	0	**1**	0	0	**1**	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Gambar 1. Arah pengecekan piksel pada array biner citra

b. Cek 8 piksel tetangga dari piksel utama, jika bernilai 1 maka simpan indeksnya ke array Temp

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	**21**	22	**23**	24	25	**26**	27	28	29
30	**31**	32	**33**	34	35	36	37	38	39
40	**41**	**42**	**43**	44	45	**46**	47	48	49
50	**51**	52	**53**	54	55	**56**	57	58	59
60	**61**	62	**63**	64	65	**66**	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Gambar 2. Indeks array biner

c. Ubah nilai 8 piksel tetangga menjadi `0` untuk menghindari pembacaan dan penambahan ulang.

d. Cek 8 piksel tetangga dari setiap indeks piksel yang tersimpan dalam array Temp.

e. Ulangi *scanning* ke semua 8 piksel tetangga dari setiap indeks piksel yang ada dalam array Temp hingga tidak ada piksel tetangga yang bernilai 1.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	**0**	0	**0**	0	0	**0**	0	0	0
0	**0**	0	**0**	0	0	0	0	0	0
0	**0**	**0**	**0**	0	0	**0**	0	0	0
0	**0**	0	**0**	0	0	**0**	0	0	0
0	**0**	0	**0**	0	0	**0**	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Gambar 3. Array biner citra setelah *scanning* selesai

f. Tambahkan semua indeks piksel pada Temp ke array CC

ISSN : 2597 - 4696

g. Kembali melakukan *scanning* piksel poin a sampai akhir dari array biner.

h. Lakukan konversi indeks piksel menjadi koordinat x,y dengan persamaan:

y = indeks / width

x = indeks – (y * width)

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	10	11	12	13	14	15	16	17	18	19
2	20	**21**	22	**23**	24	25	**26**	27	28	29
3	30	**31**	32	**33**	34	35	36	37	38	39
4	40	**41**	**42**	**43**	44	45	**46**	47	48	49
5	50	**51**	52	**53**	54	55	**56**	57	58	59
6	60	**61**	62	**63**	64	65	**66**	67	68	69
7	70	71	72	73	74	75	76	77	78	79
8	80	81	82	83	84	85	86	87	88	89
9	90	91	92	93	94	95	96	97	98	99

Gambar 4. Tiga obyek piksel yang telah ditemukan

Dengan demikian output dari array CC adalah kumpulan array Temp dimana setiap array Temp berisi indeks piksel-piksel dari satu *connected component*. Dari contoh di atas diperoleh:

- array CC indeks pertama berisi array Temp yang membentuk obyek H berisi indeks: {21, 23, 31, 33, 41, 42, 43, 51, 53, 61, 63} dengan koordinat x, y {(1,2) ,(1,3), (1,4), (1,5), (1,6), (2,4), (3,2), (3,3), (3,4), (3,5), (3,6)}
- array CC indeks kedua berisi array Temp yang membentuk titik huruf i. Berisi indeks: {26} dengan koordinat x,y {(6,2)}. Array CC indeks ketiga berisi array Temp yang membentuk garis dari huruf i. Berisi indeks {46, 56, 66} dengan koordinat x,y {(6,4), (6,5), (6,6)}.

Berikut hasil implementasi algoritma di atas.

Gambar 5. obyek teridentifikasi

CCL mengecek nilai hitam pixel pertama dengan scanline dari kiri kanan atas bawah. Apabila ditemukan nilai hitam berikutnya, maka akan dicek pixel hitam di kiri dan sebelah atas. Jika ditemukan pixel lain berwarna hitam maka masih dianggap dalam 1 region, jika tidak maka dicek lagi mendatar dari kiri ke kanan. Ulangi prosesnya hingga pemeriksaan tidak lagi menemui pixel hitam di bagian kiri atas, kiri kanan, kemudian segmentasi menjadi satu karakter terpisah.

Secara umum, connected component labeling adalah proses pemberian label yang berbeda pada setiap karakter sehingga karakter yang satu dengan yang lain dapat dipisahkan berdasarkan label yang dimilikinya. Kelebihan dari metode connected component labeling ini sendiri adalah tidak terpengaruh pada kemiringan objek sehingga masih bisa memisahkan

ISSN : 2597 - 4696

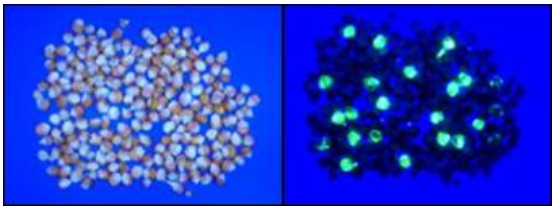
objek dengan baik walaupun posisi objek dalam image dalam keadaan miring (selama proses threshold berhasil memisahkan objek dengan jelas). Proses atau tahapan CCL ini akan dipergunakan baik dalam mendeteksi dan menghitung jagung maupun aflatoksin. Hasil dari deteksi melalui CCL tersebut yaitu setiap obyek baik jagung maupun aflatoksin akan diberi warna yang berbeda sebagai keberhasilan identifikasi yang telah dilakukan oleh sistem.

METODE PENELITIAN

1. Pengambilan citra

Pengambilan citra jagung UV menggunakan kamera berfitur WIFI yang dimasukkan ke dalam box percobaan sebagai berikut.

Gambar 6. Kotak uji coba



Gambar 7. Pengambilan citra normal dan UV

2. Thresholding untuk segmentasi warna aflatoksin.

Tujuan dari tahap ini adalah untuk mengambil warna aflatoksin dari citra UV dimana aflatoksin memiliki piksel dengan tingkat kecerahan yang lebih besar. Berikut adalah contoh hasil threshold untuk mendapatkan citra aflatoksin. Algoritmanya adalah:

Step 1 : Input image UV

Step 2 : Ambil setiap piksel dari citra untuk dikonversi menjadi piksel biner

Step 3 : Atur threshold

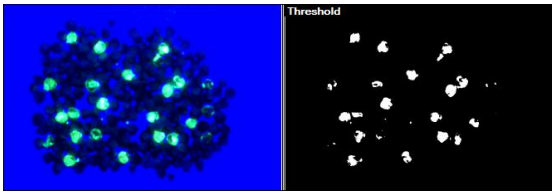
Step 4 : Ubah piksel yang bernilai 0 tetap 0, selain itu 255.

Step 5 : Ubah nilai r, g, b yang diperoleh menjadi nilai piksel

Step 6 : Ubah piksel awal dari posisi i,j menjadi nilai variabel untuk koordinat piksel tersebut

Step 7 : Lanjutkan pengulangan cek piksel ke seluruh koordinat piksel hingga piksel yang terakhir pada citra

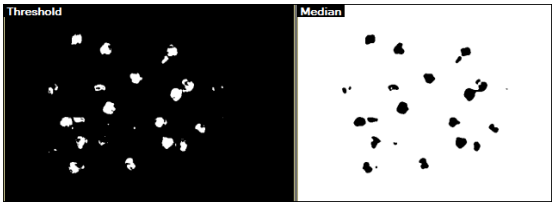
ISSN : 2597 - 4696



Gambar 8. Threshold menghasilkan piksel aflatoksin

3. Median filter

Dari thresholding dari citra jagung UV tersebut telah diperoleh citra aflatoksin dalam nilai piksel biner. Namun dari hasil threshold tersebut ternyata masih memiliki banyak noise. Bila nilai threshold dinaikkan untuk menghilangkan noise maka bisa jadi fitur aflatoksin dapat hilang. Oleh karena itu perlu dilakukan filter median. Hal ini diperlukan untuk lebih meningkatkan akurasi dari deteksi aflatoksin yang juga bertujuan untuk menghitung jumlah jagung yang tercermar oleh aflatoksin untuk dihitung prosentase cemarannya.

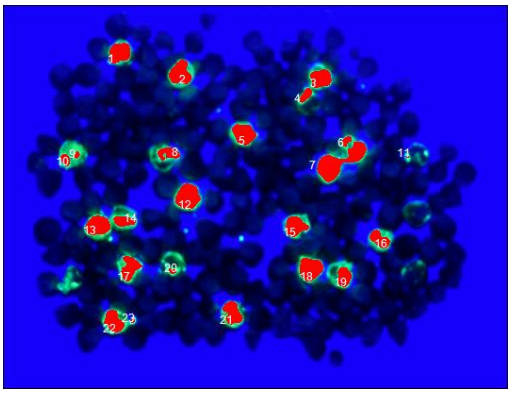


(a) (b)

Gambar 9. Citra aflatoksin (a) Noise pada hasil thresholding. (b) Hasil filter median

4. Perhitungan obyek aflatoksin

Karena yang dihitung adalah jumlah sekaligus luasan dari aflatoksin untuk mengetahui prosentasenya nanti, maka tahapan CCL pada obyek aflatoksin tidak menggunakan piksel tepi, namun dilakukan dengan menghitung seluruh piksel dalam obyek tersebut dan menghitung total jumlah obyek yang ditemukan sebagai aflatoksin. Jumlah piksel dari semua obyek aflatoksin akan dihitung total untuk dibandingkan dengan total jumlah piksel citra median pada jagung dengan tujuan untuk mengetahui prosentase luasan citra aflatoksin terhadap luasan citra seluruh jagung.



Gambar 9. Identifikasi aflatoksin

5. Prediksi Jumlah Jagung

Berikut implementasi dari tahap-tahap dalam prediksi jumlah jagung.

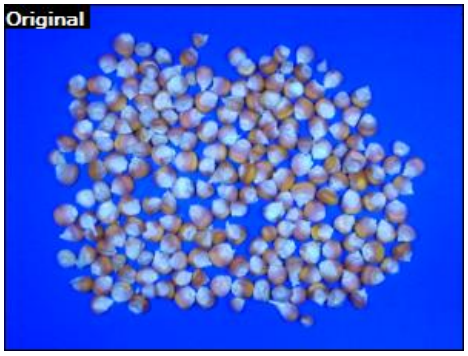
a. Simplified Gabor Wavelet

Pada tahap ini, akan diterapkan pada citra jagung non UV. Menggunakan fungsi gabor wavelet yang menggunakan imaginary part dari filter gabor:

ISSN : 2597 - 4696

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = \exp\left(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2}\right)\sin\left(2\pi\frac{x'}{\lambda}+\psi\right)$$

Berikut adalah contoh implementasi langsung pada citra di bawah ini.

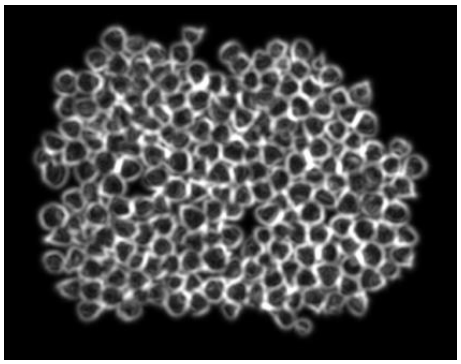


Gambar 10. Citra jagung non UV

Parameter yang diberikan adalah lambda = 5, theta = 0, psi = [0 pi/2], gamma = 0.5, bw = 1, N = 8.
akan menghasilkan 8 sudut orientasi dengan fungsi gabor sebagai berikut.

Gambar 11. Citra mask filter gabor 8 orientasi dimana θ=θ+pi/8

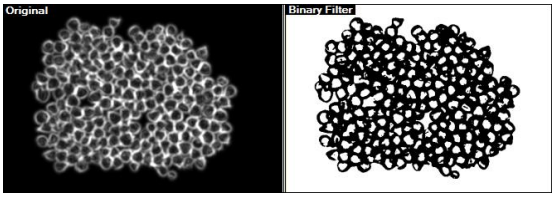
Selanjutnya kernel ini akan dikonvolusikan pada citra jagung di atas sehingga akan menghasilkan citra SGW sebagai berikut.



Gambar 12. Citra hasil SGW 8 orientasi dimana θ=θ+pi/8

b. Thresholding

Langkah selanjutnya adalah melakukan double thresholding untuk mengidentifikasi obyek jagung pipilan yang sudah tersegmentasi pada citra hasil SGW. Selain itu pada tahap thresholding ini juga akan membersihkan piksel-piksel yang tidak diperlukan. Berikut adalah citra hasil proses double thresholding dimana batasan antara obyek pipilan jagung semakin jelas. Namun masih ditemukan noise yaitu berupa obyek-obyek kecil yang bukan obyek utama.

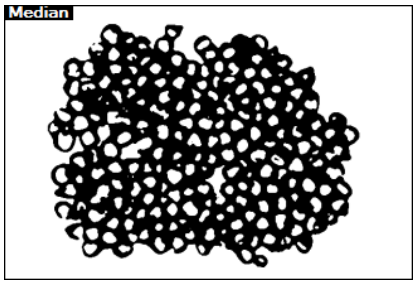


Gambar 10. Hasil implementasi double thresholding pada SGW

ISSN : 2597 - 4696

c. Median

Untuk mengurangi noise tersebut perlu dilakukan filter median dengan contoh filter median 3x3.



Gambar 11. Citra filter median 3x3, 5 iterasi

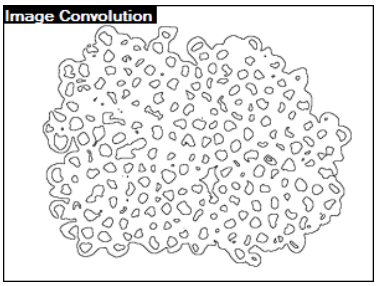
Semakin tinggi nilai iterasi yang diterapkan maka noise juga semakin berkurang, namun akan terdapat piksel obyek yang saling menyambung dengan piksel obyek yang berdekatan. Sehingga pada proses perhitungan akan di cek kembali piksel-piksel yang terlalu kecil tidak akan dihitung sebagai obyek

d. Image Convolution

Citra hasil dari filter median disimpan dan akan dikonvolusikan kernel untuk deteksi tepi. Berikut adalah hasil konvolusi citra filter median dengan kernel 3x3 operator Laplacian.

$$\begin{bmatrix} 1 & 1 & -1 \\ 1 & -8 & 1 \\ 1 & 1 & -1 \end{bmatrix}$$

Gambar 12. Operator Laplacian (termasuk diagonal) untuk deteksi tepi



Gambar 13. Hasil konvolusi citra dengan kernel 3x3 operator Laplacian

e. Connected Component Labeling

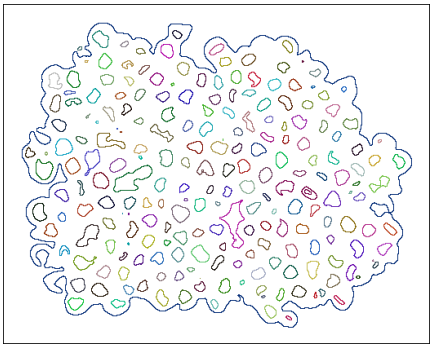
Langkah selanjutnya adalah melakukan perhitungan obyek jagung dengan mengindentifikasi piksel tertutup yang membentuk obyek jagung dengan menggunakan CCL. Mendeteksi obyek dengan menggunakan tepi lebih menghemat waktu komputasi dari pada menggunakan piksel pada obyek secara keseluruhan bidang obyek. Untuk mendeteksi piksel dari 8 tetangga menggunakan indeks piksel dapat menggunakan persamaan berikut.

i-width-1	i-width	i-width+1
i-1	i	i+1
i+width-1	i+width	i+width+1

Gambar 14. Persamaan menentukan indeks 8-connected pixels

Berikut adalah hasil tahapan CCL pada citra konvolusi deteksi tepi diinputkan.

ISSN : 2597 - 4696



Gambar 15. Citra hasil CCL

6. Hitung Prosentase Cemaran Aflatoksin Terhadap Jagung

Untuk mengetahui prosentase cemaran aflatoksin terhadap jagung maka harus diketahui dulu pembandingnya dan luasan piksel aflatoksin. Luasan aflatoksin dapat diketahui dengan menghitung jumlah piksel dalam obyek tersebut untuk semua obyek aflatoksin yang ditemukan.

Bila telah diketahui jumlah piksel jagung dan jumlah piksel jamur maka dapat dihitung prosentase luasan atau cemaran atau kandungan aflatoksin terhadap data jagung sebagai berikut.

Prosentase Luasan =
(JumlahPixelPenuhdariJamur /
..JumlahPixelPenuh Keseluruhan Jagung
)*100%

Setelah proses median filter, maka dapat dilakukan perhitungan obyek dengan algoritma Connected Component Library. Untuk selanjutnya citra hasil hitung ini dipergunakan untuk menghitung prosentase antara jumlah Aflatoksin dari keseluruhan jumlah jagung. Sedangkan untuk menghitung prosentase luasan Aflatoksin terhadap jagung, digunakan perbandingan jumlah total piksel Aflatoksin terhadap jumlah total piksel jagung dari citra hasil median filter tahap hitung jagung.

HASIL PENELITIAN DAN PEMBAHASAN

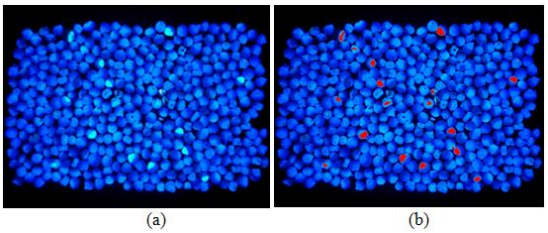
Pada tahap pengujian tersebut, dapat digunakan minimum dan maksimum *thresholding* dan variasi iterasi untuk filter median digunakan untuk menyesuaikan kondisi posisi jagung pada citra secara keseluruhan. Hal ini dimaksudkan agar perhitungan dapat lebih akurat. Rata-rata nilai minimum thresholding adalah 44 hingga 50 dan maksimum 255 untuk mendapatkan obyek biner jagung dari citra UV.

Sedang iterasi filter median menggunakan nilai 4. Sedangkan untuk *thresholding* deteksi aflatoksin rata-rata menggunakan nilai *threshold* 150 dan 5 iterasi filter median. Dari 100 citra tersebut dapat dihasilkan prosentase akurasi prediksi aflatoksin bervariasi antara 70% hingga 100%. Hal ini disebabkan pada data uji terdapat aflatoksin yang posisinya hanya ada

ISSN : 2597 - 4696

di salah satu sisi bulir jagung. Sehingga pada waktu posisi bulir jagung yang mengandung aflatoksin diubah, maka ada kemungkin aflatoksin tidak tampak. Sedangkan prosentase akurasi prediksi jumlah jagung menggunakan SGW bervariasi antara 89% hingga 100% dengan jumlah total jagung 390 bulir dengan rata-rata akurasi dari 100 citra adalah 95% dengan rata-rata peroleh jumlah jagung 369 buah.

Rata-rata jumlah piksel aflatoksin untuk citra dengan 20 aflatoksin adalah 2201 piksel. Sedangkan rata-rata jumlah piksel aflatoksin untuk citra dengan 10 aflatoksin adalah 739 piksel. Dan rata-rata jumlah total piksel penuh jagung atau luasannya adalah 205.575 (resolusi citra 640x480) sehingga dapat diperoleh prosentase rata-rata cemaran aflatoksin adalah sebesar 0.43%. Nilai yang sama juga diperoleh bila menghitung rata-rata dari prosentase prediksi cemaran berdasarkan luasan (*area based*) dari seluruh dataset yaitu 0.4%. Dari hasil uji juga diperoleh bahwa ukuran tiap obyek jagung yang diuji ternyata bervariasi hanya pada nilai 50 hingga 70 piksel, sehingga rata-rata ukuran obyek jagung adalah 64 piksel.



Gambar 16. (a) Citra asli UV dengan 20 afltoksin (b) Citra aflatoksin ditandai piksel warna merah

Dari pengujian 10 citra dengan 20 aflatoksin dan 90 citra dengan 10 afltoksin serta total jagung 390 rata-rata ditemukan 19 buah atau 95%. Sedangkan dari pengujian 90 citra dengan 10 aflatoksin dan total jagung 390 rata-rata ditemukan 8,8 dibulatkan 9 buah atau 90%.

KESIMPULAN DAN SARAN

Dari hasil penelitian ini maka dapat ditarik beberapa kesimpulan sebagai berikut :

a. Perhitungan jumlah biji jagung harus menggunakan citra jagung non UV. Hal ini disebabkan pada citra UV fitur tepi dan fitur lain seperti warna yang membedakan antara butir jagung satu dengan yang lainnya sudah berkurang bahkan hilang sehingga mengurangi nilai akurasi perhitungan.

b. Algoritma CCL dapat digunakan secara optimal untuk menghitung obyek biji jagung, maka citra harus melalui tahap binary filter,

ISSN : 2597 - 4696

median filter, serta image convolution terlebih dahulu.

REFERENSI

Wing-Pong Choi, Siu-Hong Tse, Kwok-Wai Wong & Kin-Man Lam. (2008). Simplified Gabor wavelets for human face recognition, Elsevier, Pattern Recognition 41, pp. 1186-1199

Wei Jiang, Kin-Man Lam & Ting-Zhi Shen. (2009). Efficient Edge Detection Using Simplified Gabor Wavelets, IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics, Vol.39, No.4, pp.1036-1047.

Wei Jiang, Ting-Zhi Shen, Yu Hu & Xin-Yi Wang. (2008). Gabor wavelets for Image Processing, IEEE International Colloquium on Computing, Communication, Control, and Management, Vol.1.pp.110—114.

Selvathi, D., Sujatha, C. (2012). An Optimal Solution For Image Edge Detection Problem Using Simplified Gabor Wavelet, International Journal of Computer Science, Engineering and Information Technology, Vol.2, No.3, pp. 99-115.

Song-Iin Liu, Zhao-dong Niu, Gang Sun, & Zeng-ping Chen. (2014). Gabor Filter-Base Edge Detection: A Note, Elsevier, Optik 125, pp.4120-4123.

Serrano, A., Diego, I.M., Conde, C., & Cabello, E. (2010). Recent Advances in Face Biometrics with Gabor Wavelets, Elsevier Pattern Recognition Letters, Vol. 31, pp. 372-381.

Tai Sing Lee. (1996). Image Representation Using 2D Gabor Wavelets, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol 18, No. 10.

Grigorescu, S.E., Petkov, N., & Kruizinga, P. (2002). Comparison of Texture Features Based on Gabor Filters, IEEE Transactions on Image Processing, Vol. 11, Nol. 10, pp. 1160-1167.

Yiming Ji, Kai H. Chang, & Chi-Cheng Hung. (2004). Efficient Edge Detection and Object Segmentation Using Gabor Filters, ACMSE, pp. 454-459.

Jian-Jun Hao, Qiang Jiang, Jian-Wei Wei, & Lin Mi. (2010) Research of Edge Detection Based on Gabor Wavelet, IEEE International Conference on Measuring Technology and Mechatronics Automation, Vol. 2, pp.1083-1086.

Gonzalez. R & Woods R.E. (1992). Digital Image Processing, Addision- Wesley Publishing Co.Inc.

Jun Li. (2003). A Wavelet Approach to Edge Detection. Thesis The Department of Mathematics and Statistics Sam Houston State University.

Daugman, J.G. (1988). Complete Discrete 2-D Gabor Transforms by Neural Network for Image Analysis and Compression, IEEE Transaction On Acoustics, Speech, And Signal Processing, Vol. 36, No. 7, pp. 1169-1179.

Fu Chang, Chun-Jen Chen, & Chi-Jen Lu. (2003). A Linear-time Component-Labeling Algorithm Using Contour Tracing Technique, Elsevier, Computer Vision and Image Understanding

 ISSN : 2597 - 4696