



Model Identifikasi Singkong Berdasarkan Warna untuk Tepung Mocaf Berbasis Citra Digital

Sri Andayani^{1*} , Ega Noviasuti¹

¹Program Studi Matematika, Universitas Negeri Yogyakarta, Indonesia

* Corresponding Author. E-mail: andayani@uny.ac.id

ARTICLE INFO

Article History:

Received: 09-Oct. 2020

Revised: 07-Jun. 2022

Accepted: 18-Sep. 2022

Keywords:

Citra digital, segmentasi threshold, binerisasi, singkong, mocaf

ABSTRACT

Penelitian ini bertujuan untuk menyusun model untuk mengidentifikasi mutu singkong berdasarkan warna sebagai bahan pembuatan tepung mocaf dengan berbasis citra digital. Metode yang digunakan meliputi beberapa tahap pengolahan citra digital antara lain preprocessing dan ekstraksi ciri. Preprocessing meliputi cropping, resizing, dan grayscale, sedangkan ekstraksi ciri meliputi segmentasi threshold dan binerisasi. Data penelitian menggunakan 188 citra singkong yang dibagi menjadi 72 citra data latih dan 46 data uji. Penelitian memberikan hasil berupa model identifikasi berdasarkan 2 hal berikut: a) menggunakan ekstraksi ciri yang meliputi segmentasi threshold dengan nilai ambang 170 dan binerisasi dengan nilai ambang 75; dan b) penentuan mutu singkong dilakukan berdasarkan perbandingan luas piksel putih hasil segmentasi threshold dengan luas piksel putih hasil binerisasi. Singkong dikatakan bermutu baik jika citranya yang memiliki persentase luas piksel warna putih $\geq 65\%$. Model yang dihasilkan memberikan akurasi sebesar 94% terhadap 72 data latih dan sebesar 95% terhadap 46 data uji.

This study aims to develop a model to identify the quality of cassava based on color as an ingredient for making mocaf flour based on digital images. The procedure includes preprocessing and feature extraction among other steps of digital image processing. Preprocessing includes cropping, resizing, and grayscale, while feature extraction includes threshold segmentation and binaryzation. The data are 188 cassava images consisting of 72 training data images and 46 test data. The research yields a model based on the following 2 things: a) utilizing feature extraction that uses threshold segmentation with a threshold value of 170 and binaryzation with a threshold value of 75; and b) determining of the quality of cassava is carried out based on the ratio of the area of white pixels produced by threshold segmentation to the area of white pixels produced by binaryzation. If 65% or more of the pixels in the image are white, the cassava is considered to be of good quality. The resulting model provides an accuracy of 94% for 72 training data and 95% for 46 test data.




This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-nc-sa/4.0/) license



How to Cite:

Andayani, S., & Noviasuti, E. (2022). Model identifikasi singkong berdasarkan warna untuk tepung mocaf berbasis citra digital. *Pythagoras: Jurnal Matematika dan Pendidikan Matematika*, 17(1), 281-292. <https://doi.org/10.21831/pythagoras.v17i1.34994>

 <https://doi.org/10.21831/pythagoras.v17i1.34994>

PENDAHULUAN

Aplikasi teknologi komputer yang berkembang pesat di berbagai bidang memberikan banyak manfaat dalam meningkatkan kualitas kehidupan manusia. Salah satu penerapannya adalah pengolahan citra digital, yaitu cabang ilmu komputer yang mempelajari hal-hal yang berkaitan dengan gambar seperti perbaikan kualitas gambar, transformasi gambar, dan lain-lain (Sutoyo et al., 2009). Banyak penelitian dilakukan terkait dengan penggunaan citra digital dalam identifikasi bagian dari tanaman seperti pengenalan daun atau buah. Salah satunya, adalah penggunaan segmentasi citra dengan metode *threshold* pada citra tanaman narkotika dengan hasil penelitian yaitu nilai ambang (T) 0,7 untuk pengenalan daun ganja, nilai ambang (T) 0,4 untuk pengenalan khant dan nilai ambang

0,5 untuk pengenalan koka ([Angriani et al., 2015](#)). Contoh lainnya yaitu analisis data citra buah-buahan dengan algoritma *fagin* dan *threshold* dengan hasil nilai presisi yang optimal adalah 92% untuk algoritma *fagin* dan 94% untuk algoritma *threshold* ([Rangkuti, 2007](#)).

Selain untuk analisis citra, pengolahan citra digital juga digunakan untuk pengidentifikasi atau klasifikasi. Pengklasifikasian kematangan buah yang dilakukan pada produksi rumahan biasanya menggunakan cara manual yaitu dengan cara pengamatan visual secara langsung dengan mengamati buah yang diidentifikasi. Hal tersebut akan membutuhkan biaya tenaga kerja dan dapat memiliki hasil yang berbeda-beda karena subjektivitas operator sortir sehingga hasil pengamatannya tidak konsisten ([Nanda et al., 2018](#)). Oleh sebab itu, pengolahan citra digital banyak ditawarkan sebagai alternatif solusi untuk pengidentifikasi produk pertanian dan perkebunan secara otomatis termasuk untuk melakukan klasifikasi ([Kusumaningtyas & Asmara, 2016](#)).

Salah satu contoh penelitian klasifikasi jenis dan kematangan buah tomat berdasarkan bentuk dan ukuran serta warna permukaan kulit buah berbasis pengolahan citra digital dengan tingkat akurasi klasifikasi bentuk sebesar 86,17% dengan nilai ambang sebesar 0,55, akurasi klasifikasi ukuran sebesar 84,04% dengan nilai ambang sebesar 0,5, akurasi klasifikasi kematangan sebesar 80,85% dengan nilai ambang sebesar 0,7 dan akurasi keseluruhan sebesar 54,26% dengan nilai ambang sebesar 0,75 ([Saraswati et al., 2010](#)). Penelitian lainnya adalah pengembangan aplikasi pemilihan buah tomat untuk bibit unggul berdasarkan warna dan ukuran menggunakan *hue saturation value* (HSV) dan nilai ambang dengan hasil yaitu aplikasi yang dibuat mampu menampilkan kualitas warna maupun ukuran dari buah tomat berupa kualitas bagus, setengah bagus, dan tidak bagus, dapat menampilkan nilai warna dari buah tomat berupa nilai *hue, saturation, value*, dan kuantitas HSV, mampu menampilkan nilai ukuran dari buah tomat berupa nilai banyak piksel objek ([Maula et al., 2016](#)).

Pada penelitian ini, objek yang akan diteliti adalah singkong atau ubi kayu. Singkong merupakan produk pertanian yang cocok dikembangkan di segala jenis tanah di Indonesia. Selain cocok dikembangkan di Indonesia, ubi singkong dapat dimanfaatkan sebagai alternatif pangan pengganti beras maupun gandum karena karbohidrat yang terkandung dalam ubi singkong sangat tinggi. Oleh sebab itu, singkong merupakan salah satu bahan yang prospektif dikembangkan sebagai bahan diversifikasi pangan. Singkong dapat diolah menjadi berbagai macam olahan salah satunya yaitu diolah menjadi tepung. Tepung singkong yang dapat digunakan untuk substitusi tepung terigu yaitu tepung mocaf (*modified cassava flour*). Ketersediaan bahan baku yang melimpah membuat tepung mocaf memiliki prospek yang baik ([Yani & Akbar, 2018](#)). Pemanfaatan singkong menjadi bahan baku tepung mocaf mampu mengurangi penggunaan tepung terigu/gandum ([Gusriani et al., 2021](#)). Selain itu, singkong yang diolah menjadi tepung mocaf juga dapat menjadi komoditas industri pangan ([Hasditio & Fitri, 2019](#)). Tepung mocaf lebih fleksibel apabila digunakan sebagai substitusi tepung terigu dengan proporsi 10% sampai 100% ([Yulifianti et al., 2012](#)). Tepung mocaf terhadap tepung terigu juga dapat dimanfaatkan pada produk roti basah dan mie yakni berkisar 20%-40%, tetapi untuk produk *cookies* dapat mencapai 100% sehingga tepung mocaf potensial dikembangkan untuk substitusi tepung terigu khususnya pada pembuatan produk pangan ([Asmoro, 2021](#)). Oleh sebab itu, tepung mocaf dapat menjadi alternatif atau substitusi untuk tepung terigu sehingga diharapkan dapat mengurangi tingkat impor tepung terigu.

Pembuatan tepung mocaf melalui beberapa tahap, salah satunya yaitu tahap penyortiran ubi singkong. Penyortiran ubi singkong bertujuan untuk memisahkan antara singkong tidak mengandung HCN dengan singkong yang mengandung HCN. Singkong yang digunakan adalah singkong yang aman untuk dikonsumsi yaitu singkong yang tidak mengandung HCN atau sekurang-kurangnya mengandung HCN sebanyak 20-50mg per kilo. Ciri-ciri singkong yang mengandung HCN yaitu terdapat bercak atau garis berwarna biru kehitaman pada ubi singkong ([Muticara, 2016](#)). Ciri-ciri fisik singkong yang baik dan buruk yang tampak dari adanya bercak atau garis biru kehitaman tersebut dapat digunakan sebagai dasar dalam identifikasi kualitas singkong, sehingga dapat membantu penyortiran singkong sebagai bahan baku tepung mocaf. Dengan menggunakan citranya, proses identifikasi kualitas ubi singkong dapat diotomatisasi dengan berbantuan teknologi komputer.

Beberapa aplikasi komputer dalam penelitian dengan objek singkong ditemukan, yakni dalam hal deteksi penyakit singkong melalui citra daunnya. Metode yang digunakan antara lain neural network ([Lilhore et al., 2022](#)), deep learning ([Ramcharan et al., 2017](#); [Metlek, 2021](#)), convolutional neural network ([Oyewola et al., 2021](#)), dan machine learning ([Tusubira et al., 2020](#)). Dari penelitian-penelitian tersebut, belum ditemukan penelitian yang menggunakan objek citra ubi singkong. Oleh karena ini, melalui penelitian ini, teknik pengolahan citra digital diterapkan untuk memperoleh model pengklasifikasian kualitas singkong dengan menggunakan data citra ubi

singkong. Model yang dikembangkan diharapkan dapat membantu proses penyortiran singkong sebagai bahan tepung mocaf.

METODE

Data

Penelitian ini menggunakan data primer yaitu pengumpulan data secara langsung dilakukan dengan cara memfoto atau mengambil gambar objek (singkong) secara langsung. Waktu yang digunakan untuk pengambilan data penelitian ini yaitu dalam kurun waktu 2 hari, terhitung mulai tanggal 10 Desember 2018 hingga 11 Desember 2018. Total data citra yang digunakan pada penelitian ini sebanyak 118 gambar singkong. Data citra yang digunakan untuk data latih sebanyak 72 gambar (36 gambar singkong baik dan 36 singkong buruk). Kemudian 46 gambar (17 gambar singkong baik dan 29 singkong buruk) digunakan untuk data uji. Foto singkong diambil dalam jarak 35cm dihitung dari letak objek (singkong).

Data citra singkong dikelompokkan berdasarkan fisik singkong aslinya, yakni singkong dengan kualitas baik dan kualitas buruk. Hal ini dilakukan sebagai acuan dalam menghitung tingkat akurasi program dengan cara membandingkan antara citra asli berdasarkan klasifikasi fisik singkong aslinya dengan citra singkong hasil klasifikasi menggunakan program. Apabila klasifikasi fisik dan program menghasilkan keputusan yang sama akan dihitung sebagai nilai benar dan apabila tidak sama maka akan dihitung sebagai nilai salah.

Pendekatan metode yang digunakan

Metode yang digunakan dalam penelitian ini adalah pengolahan citra, yaitu bidang ilmu yang mempelajari perbaikan citra sehingga manusia dan komputer mudah dalam menginterpretasi citra ([Hidayatullah, 2017](#)). Beberapa hal yang dipelajari dalam pengolahan citra yaitu perbaikan kualitas citra, transformasi citra, pemilihan citra ciri yang optimal untuk analisis, proses penarikan informasi data untuk penyimpanan data, transmisi data, dan waktu proses data. *Input* dalam pengolahan citra yaitu berupa citra dan *output*nya juga berupa citra yang diperoleh dari hasil pengolahan ([Sutoyo et al., 2009](#)).

Citra merupakan gambaran atau imitasi dari suatu objek. Citra digital adalah citra yang didapat oleh komputer misalnya citra yang dihasilkan oleh kamera digital dan *scanner* ([Kadir, 2013](#)). Citra digital tersusun dari sejumlah piksel (*picture element*) yang membentuk suatu matriks. Piksel adalah perpotongan antara kolom dan baris. Citra memiliki sejumlah elemen dasar yang dapat dimanipulasi dalam pengolahan citra ([Munir, 2004](#)). Elemen-elemen tersebut antara lain yaitu kecerahan, kontras, kontur, warna, bentuk, dan tekstur. Berdasarkan nilai pikselnya, citra dibagi menjadi 3 jenis yaitu citra warna, citra *grayscale*, dan citra biner.

Dalam proses pengolahan citra digital dikenal adanya segmentasi, yaitu metode untuk membagi citra menjadi beberapa daerah yang setiap daerahnya memiliki kemiripan atribut ([Putra, 2010](#)). Segmentasi membagi daerah mengikuti tingkatan yang diinginkan sesuai dengan masalah yang akan diselesaikan. Proses segmentasi akan berhenti jika pembagian daerah yang diinginkan dalam aplikasi telah terpenuhi ([Prasetyo, 2011](#)). Segmentasi citra mempunyai sifat *discontinuity* atau *similarity* dari intensitas piksel. Pendekatan *discontinuity* yaitu mempartisi citra apabila terdapat perubahan intensitas secara tiba-tiba (*edge based*). Pendekatan *similarity* yaitu mempartisi citra menjadi daerah-daerah yang memiliki kesamaan sifat tertentu (*region based*), contohnya yaitu : *thresholding*, *region growing*, *region splitting and merging* ([Sinaga, 2017](#)).

Penelitian ini menggunakan *thresholding*, yakni metode segmentasi dengan cara memilih nilai ambang atau nilai batas intensitas tertentu sehingga dapat mempartisi citra yang akan disegmentasi. Dengan kata lain, *thresholding* merupakan proses mengubah citra *grayscale* menjadi suatu citra biner ([Saputra et al., 2019](#)). Cara kerja *thresholding* yaitu dengan cara memberikan nilai 1 pada piksel yang bernilai lebih besar atau sama dengan nilai ambang (T) yang telah ditentukan dan memberikan nilai 0 pada piksel yang nilainya kurang dari nilai ambang (T). Penentuan nilai ambang (T) secara manual yaitu dengan cara *trial and error* ([Putra, 2010](#)). Hasil dari proses *thresholding* yaitu citra biner $b(x, y)$ dengan nilai piksel 1 dan 0 dari citra *grayscale* $g(x, y)$. *Thresholding* pada $b(x, y)$ dapat didefinisikan sebagai berikut :

$$b(x, y) = \begin{cases} 1 & \text{jika } g(x, y) \geq T \\ 0 & \text{jika } g(x, y) < T \end{cases}$$

Piksel yang berkaitan dengan objek diberi nilai 1 dan piksel yang berkaitan dengan *background* diberi nilai 0. *Thresholding* ada 2 jenis yaitu *global thresholding* (pengambangan secara global) dan *locally thresholding* (pengambangan secara lokal adaptif).

Pengujian Model

Model yang diperoleh diimplementasikan dengan menggunakan program. Pengujian model identifikasi singkong dilakukan dengan cara membandingkan *output* program dengan data klasifikasi fisik singkong asli. Hasil pengujian ini adalah sebuah angka persentase tingkat akurasi program. Perhitungan persentase tingkat akurasi dihitung dengan rumus sebagai berikut:

$$akurasi = \frac{\sum data\ benar}{\sum data\ uji} \times 100\%$$

Akurasi adalah suatu nilai yang digunakan untuk mengetahui tingkat keberhasilan program yang telah dibuat. Data benar adalah hasil dari pembacaan program yang benar dibandingkan dengan klasifikasi manual.

HASIL DAN PEMBAHASAN

Hasil penelitian ini berisikan penjelasan secara detail tahap-tahap pembuatan program pengolahan citra digital dalam identifikasi mutu singkong berdasarkan warna untuk tepung mocaf yang meliputi *preprocessing* citra, ekstraksi ciri, pemilihan nilai ambang segmentasi *threshold*, pelatihan program, penentuan batas mutu singkong, dan hasil pengujian program.

1. *Preprocessing* Data Citra

Preprocessing citra merupakan proses untuk menyiapkan citra asli sehingga siap digunakan dalam sistem identifikasi singkong. Proses ini meliputi tahap-tahap untuk memperbaiki kualitas citra dengan menghilangkan *noise*, memperhalus citra, mempertajam citra, pemotongan citra (*cropping*), menghilangkan *background* citra, dan mengubah citra berwarna (RGB) menjadi citra *grayscale* maupun citra biner. Dalam penelitian ini, *preprocessing* citra yang dilakukan meliputi memotong citra, *resize* citra dan mengubah citra RGB menjadi citra *grayscale*. Berikut ini adalah penjelasan tentang tahap *preprocessing* pada citra singkong yang dilakukan pada penelitian ini.

a) Pemotongan (*Cropping*)

Proses *cropping* diperlukan pada citra asli dengan cara membuang bagian citra yang tidak digunakan dalam proses identifikasi. Proses ini diperlukan untuk memperoleh citra yang hanya memuat bagian yang diperlukan sebagai *input*. Berikut ini merupakan potongan program *crop* yang digunakan.

```
rect = getrect;
J = imcrop(gambar,[rect(1), rect(2), rect(3), rect(4)]);
crop = J;
```

Perintah *getrect* digunakan untuk memilih area berupa persegi panjang dari citra yang diinginkan dengan menggunakan *mouse*. *Imcrop (gambar,[rect(1), rect(2), rect(3), rect(4)])* berfungsi untuk memotong gambar yang diinginkan.

b) *Resizing*

Pada tahap ini, hasil dari proses *cropping* akan diubah ukuran pikselnya untuk menghindari citra blur. Citra blur adalah citra yang tidak jelas atau kabur. Skala yang digunakan pada penelitian ini yaitu sebesar 0.5. Berikut ini perintah untuk proses *resizing*.

```
scale=str2double(scale);
J = imresize(I, scale);
resize = J;
```

Perintah *str2double* digunakan untuk mengkonversi teks menjadi angka. *Imresize* merupakan perintah pada matlab yang berfungsi untuk mengembalikan gambar pada ukuran tertentu.

c) Mengubah citra RGB menjadi *grayscale*

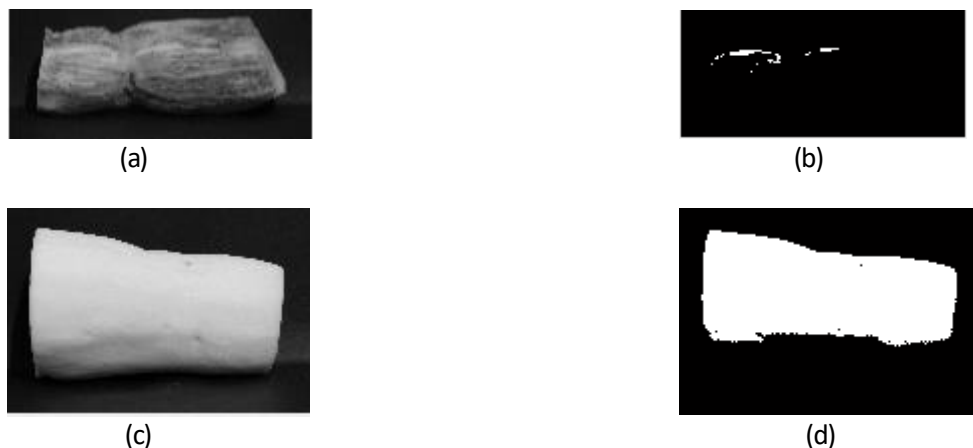
Tahap selanjutnya yaitu mengubah citra RGB menjadi *grayscale*. Proses ini diperlukan agar pengolahan citra lebih mudah karena citra *grayscale* hanya memiliki satu kanal pada pikselnya. Berikut ini adalah perintah untuk proses mengubah citra RGB menjadi *grayscale*.

```
%Mengubah cirta ke grayscale
global grayscale;
R=resize(:,:,1);
G=resize(:,:,2);
B=resize(:,:,3);
gray=0.3333*R + 0.3333*G + 0.3333*B;
grayscale = gray;
axes(handles.axes3);
imshow(gray);
```

Perintah $0,3333*R+0,3333*G+0,3333*B$ digunakan untuk mengubah citra RGB menjadi citra *grayscale*.

2. Ekstraksi Ciri

Ekstraksi ciri adalah metode pengambilan citra berdasarkan karakteristik histogram citra ([Permadi & Murinto, 2015](#)). Citra singkong dalam penelitian ini akan ada 2 proses ekstraksi ciri yaitu segmentasi *threshold* dan binerisasi. Nilai ambang untuk segmentasi *threshold* sebesar 170 dan untuk binerisasi nilai ambangnya sebesar 65. Hasil dari masing-masing proses kemudian dihitung jumlah piksel warna putihnya. Jumlah piksel putih pada hasil segmentasi diasumsikan sebagai luas singkong yang berwarna putih atau bagian singkong yang baik untuk dikonsumsi. Selanjutnya, untuk jumlah piksel warna putih dari hasil binerisasi diasumsikan sebagai luas dari seluruh citra singkong. Perbandingan antara jumlah piksel warna putih hasil segmentasi *threshold* dengan hasil binerisasi tersebut yang menjadi batasan antara kategori singkong yang baik dengan singkong yang buruk. Berikut ini merupakan citra hasil segmentasi *threshold* dari citra *grayscale* ditunjukkan pada [Gambar 1](#).

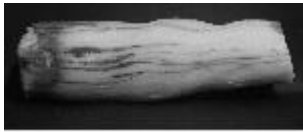






[Gambar 1](#) (a) Citra *grayscale* dari singkong buruk, (b) Hasil segmentasi *threshold* dari singkong buruk, (c) Citra *grayscale* dari singkong baik, (d) Hasil segmentasi *threshold* singkong baik.

3. Pemilihan Nilai Ambang Segmentasi *Threshold*

Segmentasi *threshold* diperlukan dalam penelitian ini adalah untuk mendapatkan bagian singkong yang dianggap putih. Pemilihan nilai ambang (T) dalam tahap segmentasi *threshold* menggunakan cara *trial and error* dengan variasi nilai ambang yaitu 100, 150, 170, dan 175. [Tabel 1](#) menunjukkan data pemilihan nilai ambang untuk program identifikasi singkong dengan menggunakan metode *trial and error*.

Tabel 1. Pemilihan Nilai Ambang

Citra RGB	Nilai ambang	Hasil segmentasi singkong
	100	
	150	
	170	
	175	

Berdasarkan hasil dari Tabel 1 dapat diketahui bahwa pada nilai ambang 170, hasil segmentasi *threshold* tersebut dapat menghilangkan atau mengubah warna selain putih pada citra asli menjadi hitam pada citra biner hasil segmentasinya, sehingga penelitian ini menggunakan nilai ambang 170.

4. Pelatihan Program

Segmentasi proses pelatihan program menggunakan data citra latih sebanyak 47 citra. Citra tersebut kemudian dilakukan proses identifikasi citra melalui beberapa tahap yaitu binerisasi, segmentasi *threshold*, perhitungan jumlah piksel putih, dan menghitung persentase dari perbandingan antara jumlah piksel putih hasil segmentasi dengan jumlah piksel warna putih hasil binerisasi.

a) Binerisasi

Citra hasil dari proses *grayscale* kemudian diubah menjadi citra biner. Proses ini bertujuan untuk mengetahui luas seluruh objek singkong. Berikut merupakan perintah untuk proses binerisasi.

```

threshold = get(handles.edit2, 'String');
threshold = str2double(threshold);

%Merubah Citra Grayscale ke Biner
for f =1:m
    for e =1:n
        if (grayscale(f,e) > 75)
            Z(f,e)=255;
        else
            Z(f,e)=0;
        end
    end
end
axes(handles.axes6);
imshow(Z);
    
```

Perintah tersebut mengubah citra *grayscale* menjadi citra biner. Nilai piksel pada citra *grayscale* yang melebihi 75, maka piksel tersebut berubah menjadi putih atau nilai piksel menjadi 255. Nilai piksel yang kurang dari 75 akan berubah menjadi hitam atau nilai piksel menjadi 0. Nilai 75 ini diperoleh melalui *trial and error*. Gambar 5 adalah hasil dari tahap binerisasi.

Command Window									
27	27	30	29	29	30	31			
27	28	29	29	30	32	32			
27	29	29	29	31	32	31			
27	29	27	27	29	35	41			
27	37	56	77	98	115	134			
84	117	141	154	157	165	171			
134	162	170	170	172	173	163			
170	165	156	151	140	127	123			
157	131	124	118	130	136	131			
123	116	113	108	112	105	121			
101	97	95	106	128	142	143			
113	133	143	143	138	117	108			
128	121	112	100	95	100	118			
107	101	109	104	98	95	90			
119	99	90	92	105	111	122			
93	73	117	129	135	141	148			
69	82	114	98	108	125	138			
65	68	81	94	105	110	122			
94	89	100	102	101	111	126			
119	107	105	113	121	124	129			
121	107	100	105	114	115	121			
90	86	87	100	114	124	132			
91	82	81	84	83	91	109			
69	49	38	44	54	65	84			
56	47	45	59	73	81	93			
57	46	38	48	57	64	81			
40	27	21	28	61	71	78			

Gambar 2. Matriks grayscale



Gambar 3. Citra grayscale

Gambar 2 merupakan bentuk matriks dari citra grayscale pada Gambar 3. Matriks tersebut yang nantinya akan diubah menjadi matriks citra biner.

Command Window									
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	255	255	0	0	0	0	0
0	0	0	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	0	255	255	255	255	255
255	255	0	255	255	255	255	255	255	255
0	255	255	255	255	255	255	255	255	255
0	0	0	255	255	255	255	255	255	255
0	0	0	0	255	255	255	255	255	255
0	0	0	0	0	0	0	255	255	255
0	0	0	0	0	0	0	0	255	255
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 4. Matriks citra biner



Gambar 5. Citra biner

Setelah melakukan binerisasi maka matriks hasilnya seperti Gambar 4. Nilai piksel yang kurang dari 75 diubah menjadi 0 dan yang lebih dari 75 diubah menjadi 255. Matriks tersebut apabila berbentuk citra akan berwarna putih dan hitam seperti gambar 5. Nilai 255 pada matriks tersebut yang diolah untuk mendapatkan keputusan bahwa citra singkong tersebut termasuk singkong baik atau buruk.

b) Segmentasi *threshold*

Proses segmentasi *threshold* dalam penelitian ini berguna untuk mendapatkan data citra singkong pada bagian singkong yang dianggap baik. Jika proses binerisasi menggunakan nilai ambang 75, maka pada proses segmentasi *threshold* menggunakan nilai ambang 170. Berikut adalah perintah untuk proses segmentasi *threshold*:

```

for k=1:m
    for l=1:n
        if (grayscale(k,l) > threshold)
            G(k,l)=255;
        else
            G(k,l)=0;
        end
    end
end
axes(handles.axes4);
imshow(G);
    
```

Perintah tersebut mengubah citra *grayscale* menjadi citra biner dengan menggunakan segmentasi *threshold* yang ditunjukkan pada Gambar 6. Pixel yang nilainya lebih besar dari nilai ambang yaitu 170, maka pixel tersebut akan diubah menjadi 255 dan nilai pixel yang kurang dari 170 diubah menjadi 0, yang ditunjukkan pada Gambar 7.



Gambar 6. Citra biner hasil *threshold*

Command Window																								
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
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
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

Gambar 7. Matriks citra biner

c) Perhitungan jumlah pixel putih

Hasil dari segmentasi *threshold* dan binerisasi kemudian dihitung jumlah pixel yang berwarna putih. Berikut ini adalah perintah untuk menghitung jumlah pixel yang berwarna putih:

```
LuasZ=0;
LuasG=0;

for p=1:m
    for q=1:n

        if (G(p,q) > 0)
            LuasG=LuasG+1;
        end

        if (Z(p,q) > 0)
            LuasZ=LuasZ+1;
        end

    end
end

LuasZ
LuasG
```

Luas Z untuk menghitung pixel warna putih hasil segmentasi dan Luas G untuk menghitung pixel warna putih hasil binerisasi. Mula-mula Luas G dianggap 0. Jika nilai intensitas pada pixel tersebut lebih besar dari 0, maka Luas G ditambah 1. Hal tersebut berulang hingga pixel terakhir pada matriks. Cara yang sama juga digunakan untuk menghitung Luas Z.




d) Persentase luas

Hasil dari masing-masing Luas Z dan G kemudian digunakan untuk menentukan perbandingan dari masing-masing luas sehingga diperoleh nilai perbandingan persentase. Misalkan untuk Luas G sebesar 20 dan Luas Z sebesar 100, maka persentase luas singkong yang berwarna putih adalah $\frac{20}{100} \times 100\% = 20\%$. Angka persentase ini yang digunakan untuk menentukan kualitas atau mutu singkong.

5. Penentuan Batas Mutu Singkong

Proses penentuan angka batas mutu singkong dilakukan dengan menggunakan data latih atau *training*. Proses ini menggunakan data latih sebanyak 72 citra singkong. Tujuan dari proses ini adalah untuk menentukan angka batas atas mutu singkong yang buruk. Setiap data latih menghasilkan nilai persentase luas. Dari 72 pengamatan, terdapat nilai persentase yang kurang dari 65% dimana citra singkong tersebut mempunyai kondisi asli singkong termasuk dalam kategori buruk sehingga nilai persentase luas 65% menjadi batas atas mutu singkong buruk.

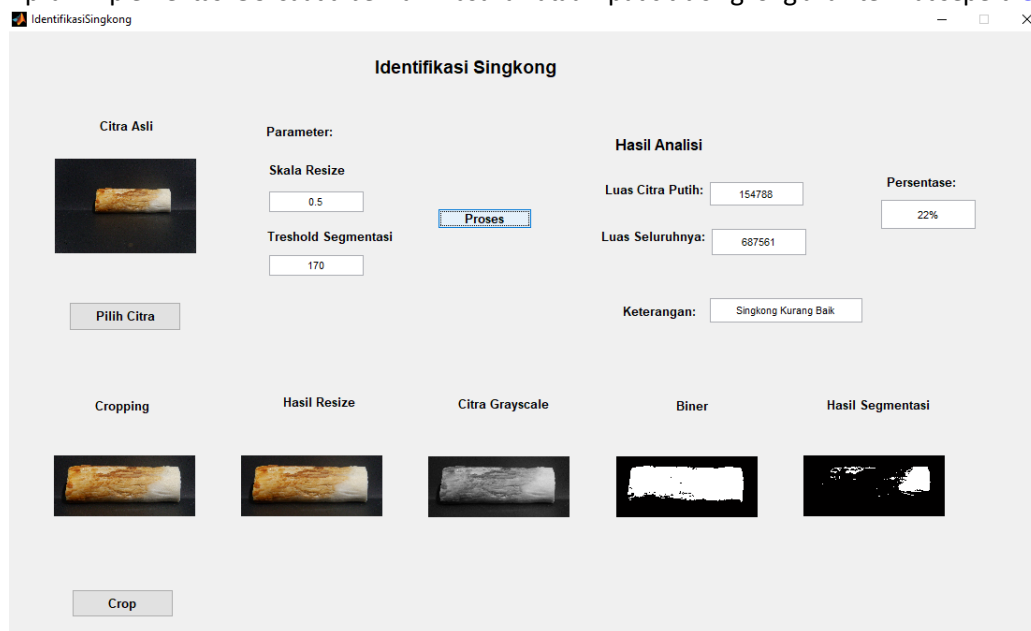
Tabel 2. Data latih

Data Ke-	Citra	Persentase Luas	Kondisi asli
1		64%	Buruk
2		65%	Baik
3		66%	Baik

Tabel 2 merupakan beberapa contoh dari data latih. Citra data apabila diidentifikasi secara keseluruhan dengan menggunakan GUI identifikasi singkong memiliki hasil akurasi sebanyak 94%.

6. Tampilan GUI Identifikasi Singkong

Tampilan implementasi GUI saat diberikan masukan atau input citra singkong akan terlihat seperti Gambar 8.



Gambar 8. GUI identifikasi singkong

Gambar 8 adalah tampilan GUI dari program identifikasi singkong yang sudah diberi *input* suatu citra singkong. Citra singkong kemudian diolah menggunakan GUI tersebut sehingga memperoleh *output* yaitu luas citra putihnya 154788, luas citra seluruhnya 687561, dan persentasenya 22%, maka citra tersebut termasuk singkong kategori buruk.

7. Hasil Pengujian Program dan Pembahasan

Pengujian sistem identifikasi singkong dilakukan pada 46 citra uji (20 singkong baik dan 26 singkong buruk) dilakukan dengan menggunakan metode segmentasi *threshold*. Tahap yang dilakukan pada pengujian adalah sama dengan tahap pelatihan tetapi menggunakan data citra uji. Sebanyak 46 citra dibedakan berdasarkan kualitas

singkongnya yaitu singkong baik dan singkong buruk. Masing-masing kualitas diuji menggunakan program yang telah dibuat, sehingga akan diketahui tingkat akurasi program yang telah dibuat. Hasil pengujian citra singkong dirangkum dalam Tabel 3.

Tabel 3. Hasil pengujian citra singkong

Jenis singkong	Jumlah citra	Jumlah citra yang terbaca benar dalam program	Jumlah error	Akurasi
Singkong baik	20	18	2	95%
Singkong buruk	26	26	0	

Berdasarkan hasil dari Tabel 3 dapat diketahui bahwa citra singkong asli yang termasuk dalam kategori baik sebanyak 20 dapat dibaca dan diidentifikasi oleh program dengan benar sebanyak 18 citra, sedangkan untuk citra asli yang termasuk kategori buruk sebanyak 26 dan dapat dibaca serta teridentifikasi secara benar oleh program sebanyak 26 citra. Hasil tersebut apabila dihitung secara keseluruhan, maka memiliki tingkat akurasi sebesar 95% yang didapatkan dari akurasi $= \frac{44}{46} \times 100\% = 95\%$.

SIMPULAN

Model yang dihasilkan untuk mengidentifikasi mutu singkong memiliki 2 prinsip berikut: a) menggunakan ekstraksi ciri yang meliputi segmentasi *threshold* dengan nilai ambang sebesar 170 dan binerisasi dengan nilai ambang sebesar 75, dan b) penentuan mutu singkong dilakukan berdasarkan perbandingan luas piksel putih hasil segmentasi *threshold* dengan luas piksel putih hasil binerisasi. Kategori untuk singkong yang baik adalah citra singkong yang memiliki persentase luas piksel warna putih $\geq 65\%$. Model yang dihasilkan memberikan akurasi sebesar 94% terhadap 72 data latih dan sebesar 95% terhadap 46 data uji.

SARAN

Hasil sistem identifikasi mutu singkong dengan menggunakan pengolahan citra digital berdasarkan warna untuk tepung mocaf masih dipengaruhi oleh kualitas citra yang diambil.

1. Dengan demikian perlu ditentukan standarisasi dalam pengambilan citra sehingga hasil identifikasi lebih bagus.
2. Perlu adanya penambahan atau perbaikan *script* GUI identifikasi singkong untuk *cropping* secara otomatis.
3. Perlu adanya perbaikan atau penambahan *script* untuk mengolah beberapa singkong dalam sekali proses.
4. Pengambilan citra singkong sebaiknya menggunakan singkong hasil panen sendiri agar mengetahui umur singkong setelah panen.

Oleh sebab itu, perlu dilakukan perbaikan pada penelitian-penelitian selanjutnya agar memperoleh hasil yang lebih baik.

DAFTAR PUSTAKA

- Angriani, L., Indrabayu, & Aren, I. S. (2015). Segmentasi citra dengan metode threshold pada citra digital tanaman narkotika. *Seminar Nasional Riset Ilmu Komputer*, 143. https://www.researchgate.net/publication/326198702_Segmentasi_Citra_dengan_Metode_Threshold_pada_Citra_Digital_Tanaman_Narkotika
- Asmoro, N. W. (2021). Karakteristik dan sifat tepung singkong termodifikasi (Mocaf) dan manfaatnya pada produk pangan. *Journal of Food and Agricultural Product*, 1(1), 34-43. <https://journal.univetbantara.ac.id/index.php/jfap/article/view/1755>
- Gusriani, I., Koto, H., & Dany, Y. (2021). Aplikasi pemanfaatan tepung mocaf (modified cassava flour) pada beberapa produk pangan di Madrasah Aliyah Mambaul Ulum Kabupaten Bengkulu Tengah. *Jurnal Inovasi Pengabdian Masyarakat Pendidikan*, 2(1), 57-73. <https://doi.org/10.33369/jurnalinovasi.v2i1.19142>

- Hasditio, A., & Fitri, S. (2019). Tepung mocaf (modified cassava flour) untuk ketahanan pangan Indonesia. *Jurnal Ilmiah Pangan Halal*, 1(1), 13-17. <https://doi.org/10.30997/jiph.v1i1.2005>
- Hidayatullah, P. (2017). *Pengolahan Citra Digital*. Bandung: Informatika.
- Kadir, A. (2013). *Dasar Pengolahan Citra dengan Delphi*. Yogyakarta: Andi Offset.
- Kusumaningtyas, S., & Asmara, R. A. (2016). Identifikasi kematangan buah tomat berdasarkan warna menggunakan metode jaringan syaraf tiruan (JST). *Jurnal Informatika Polinema*, 2(2), 72-. <https://doi.org/10.33795/jip.v2i2.59>
- Lilhore, U. K., Imoize, A. L., Lee, C.-C., Simaiya, S., Pani, S. K., Goyal, N., Kumar, A., & Li, C.-T. (2022). Enhanced convolutional neural network model for cassava leaf disease identification and classification. *Mathematics*, 10(4), 580. <https://doi.org/10.3390/math10040580>
- Maula, A. Z., Rahmad, C., & Rosiani, D. U. (2016). Pengembangan aplikasi pemilihan buah tomat untuk bibit unggul berdasarkan warna dan ukuran menggunakan HSV dan threshold. *Jurnal Teknologi Informasi*, 7(2), 127-138. <http://ejurnal.stimata.ac.id/index.php?journal=TI&page=article&op=view&path%5B%5D=223>
- Metlek, S. (2021). Disease detection from cassava leaf images with deep learning methods in web environment. *International Journal of 3D Printing Technologies and Digital Industry*, 5(3): 625-644. <https://doi.org/10.46519/ij3dptdi.1029357>
- Munir, R. (2004). *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika.
- Mutiara, N. (2016). *Pengetahuan Bahan Pangan Nabati*. Yogyakarta: Plantaxia.
- Nanda, T. R., Zulhelmi, Z., & Syaryadhi, M. (2018). Perancangan sistem sortir buah kopi berdasarkan warna dengan teknik citra digital berbasis mikrokontroler Atmega 328p. *KITEKRO: Jurnal Komputer, Informasi Teknologi, dan Elektro*, 3(2), 76-83. <https://jurnal.unsyiah.ac.id/kitekro/article/view/11401/9217>
- Oyewola, D. O., Dada, E. G., Misra, S., & Damaševičius, R. (2021). Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing. *PeerJ Computer Science*, 7:e352. <https://doi.org/10.7717/peerj-cs.352>
- Permadi, Y., & Murinto, M. (2015). Aplikasi pengolahan citra digital untuk identifikasi kematangan mentimun berdasarkan tekstur kulit buah menggunakan metode ekstraksi ciri statistik. *Jurnal Informatika*, 9(1), 1028-1038. <http://dx.doi.org/10.26555/jifo.v9i1.a2044>
- Prasetyo, E. (2011). *Pengolahan Citra Digital dan Aplikasinya Menggunakan Matlab*. Yogyakarta: Andi Offset.
- Putra, D. (2010). *Pengolahan Citra Digital*. Yogyakarta: ANDI Yogyakarta.
- Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D. P. (2017). Deep learning for image-based cassava disease detection. *Frontiers in Plant Sciences*, 8:1852. <https://doi.org/10.3389/fpls.2017.01852>
- Saputra, D. E., Rahmawati, D., & Ibadillah, A. F. (2019). Pengolahan citra digital dalam penentuan panen jamur tiram. *Jurnal Teknik Elektro dan Komputer TRIAC*, 6(1). <https://doi.org/10.21107/triac.v6i1.4356>
- Saraswati, Y., Usman, K., & Utari, W. (2010). *Sistem Klasifikasi Jenis dan Kematangan Buah Tomat Berdasarkan Bentuk dan Ukuran Serta Warna Permukaan Kulit Buah Berbasis Pengolahan Citra Digital*. Telkom University: Teknik Telekomunikasi, Fakultas Teknik Elektro. <https://openlibrary.telkomuniversity.ac.id/pustaka/92942/sistem-klasifikasi-jenis-dan-kematangan-buah-tomat-berdasarkan-bentuk-dan-ukuran-serta-warna-permukaan-kulit-buah-berbasis-pengolahan-citra-digital.html>
- Sinaga, A. S. R. M. (2017). Implementasi Teknik Thresholding Pada Segmentasi Citra Digital. *Jurnal Manajemen dan Informatika Pelita Nusantara*, 1(2), 48-51. <http://download.garuda.kemdikbud.go.id/article.php?article=746717&val=11774&title=Implementasi%20Teknik%20Thresholding%20Pada%20Segmentasi%20Citra%20Digital>

- Sutoyo, T., Mulyanto, E., Suhartono, V., Nurhayati, O. D., & Wijanarto, W. (2009). *Teori Pengolahan Citra Digital*. Yogyakarta: C.V ANDI OFFSET.
- Tusubira, J. F., Nsumba, S., Ninsiima, F., Akera, B., Acellam, G., & Nakatumba, J. (2020). Improving in-field cassava whitefly pest surveillance with machine learning. *Proceeding of Conference: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 303-309. <https://ieeexplore.ieee.org/document/9150645>
- Yani, A. V., & Akbar, M. (2018). Pembuatan tepung mocaf (modified cassava flour) dengan berbagai varietas ubi kayu dan lama fermentasi. *Edible: Jurnal Penelitian Ilmu-Ilmu Teknologi Pangan*, 7(1), 40-48 <https://doi.org/10.32502/jedb.v7i1.1655>
- Yulifianti, R., Ginting, E., & Utomo, J. S. (2012). Tepung kasava sebagai bahan substitusi terigu mendukung diversifikasi pangan. *Buletin Palawija*, 23, 1-12. <https://ejurnal.litbang.pertanian.go.id/index.php/bulpa/article/view/1315>