

DETEKSI TINGKAT KEMATANGAN BUAH MELINJO MENGUNAKAN METODE ALGORITMA *SELF ORGANIZING MAP*

Dina Estining Tyas Lufianawati^{*}), Siti Nurfa Baharani dan Rian Fahrizal

Program Studi Teknik Elektro, Universitas Sultan Ageng Tirtayasa Banten
Jl. Jendral Soedirman KM. 3 Cilegon, Banten 42435, Indonesia

^{*}E-mail: dina.lufianawati@untirta.ac.id

Abstrak

Melinjo tersebar luas di Indonesia. Buah melinjo mengalami perubahan warna dari mentah hingga sangat matang. Proses pemilihan hasil panen baik oleh petani, masyarakat, dan pelaku usaha UMKM buah melinjo pada umumnya memilih melinjo sebagai bahan dasar suatu produk masih dilakukan secara manual menggunakan penglihatan manusia normal sehingga memiliki kelemahan antara lain waktu yang dibutuhkan relatif lama dan menghasilkan produk beragam karena tingkat kelelahan manusia. Perkembangan teknologi ilmu pengetahuan dan teknologi pengolahan citra digital bisa diterapkan untuk memilih buah dari hasil panen secara otomatis dengan bantuan aplikasi pengolah citra. Tujuan penelitian ini adalah membangun sebuah sistem untuk mengklasifikasi tingkat kematangan buah melinjo dengan menggunakan metode Algoritma *Self Organizing Map* dan ekstraksi fitur RGB (*Red, Green, Blue*) dan HSV (*Hue, Saturation, Value*) dengan bantuan aplikasi pengolah citra MATLAB. Deteksi tingkat kematangan buah melinjo memiliki 4 tingkat kematangan yaitu mentah, setengah matang, matang, dan sangat matang. Penelitian ini menggunakan 200 citra, 80% data *training* dan 20% data *testing*. Berdasarkan hasil pengujian diperoleh tingkat *accuracy* 97,5% kemudian *specificity* 99,16% dan *sensitifity* 97,5%.

Kata kunci: melinjo, RGB, HSV, SOM, MATLAB

Abstract

Melinjo is widespread in Indonesia. Melinjo fruit changes colour from unripe to very ripe. The process of selecting harvests by farmers, the community, and melinjo fruit UMKM business actors generally chooses melinjo as the basic ingredient for a product is still done manually using normal human vision, so it has weaknesses, including the relatively long time required and producing various products due to the level of human fatigue. The development of scientific technology and digital image processing technology can be applied to select fruit from the harvest automatically with the help of image processing applications. The aim of this research is to build a system to classify the level of ripeness of melinjo fruit using the Self Organizing Map Algorithm method and extracting RGB (Red, Green, Blue) and HSV (Hue, Saturation, Value) features with the help of the MATLAB image processing application. Detection of the maturity level of melinjo fruit has 4 levels of maturity, namely raw, semi-ripe, ripe and very ripe. This research uses 200 images, 80% training data and 20% testing data. Based on the test results, an accuracy level of 97.5% was obtained, then specificity was 99.16% and sensitivity was 97.5%.

Keywords: melinjo, RGB, HSV, SOM, MATLAB

1. Pendahuluan

Gnetum gnemon atau yang lebih dikenal dengan nama melinjo yang berasal dari Asia Tropik dan Pasifik Barat [1] tersebar luas di Indonesia, terutama di daerah Andaman, Sumatra, dan Pulau Jawa [2]. Hampir semua bagian pohon melinjo dapat dimanfaatkan. Salah satunya adalah biji melinjo. Biji melinjo dapat dikonsumsi mentah maupun matang [3]. Salah satu jenis olahan dari biji melinjo adalah emping.

Kualitas biji melinjo sangat menentukan emping yang dihasilkan. Jika dilihat dari warna, biji melinjo yang berkualitas baik biasanya sudah benar-benar tua dan berwarna merah tua [4].

Dalam proses pemilihan buah melinjo sebagai bahan dasar pembuatan emping oleh pelaku UMKM (Usaha Mikro, Kecil, dan Menengah) dilakukan secara manual menggunakan penglihatan manusia normal sehingga memiliki kelemahan yaitu waktu yang dibutuhkan relatif lama serta menghasilkan produk yang beragam karena tingkat kelelahan dan perbedaan persepsi tentang mutu

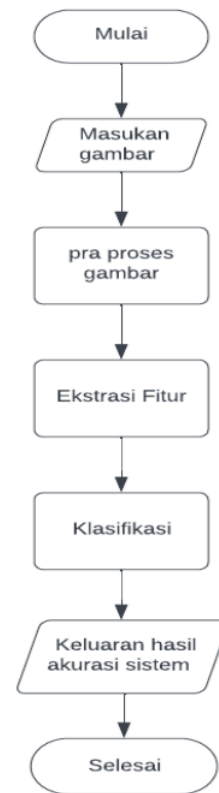
buah [1][5][6]. Salah satu indikator untuk menentukan kematangan buah melinjo yang dipakai adalah warna buah yang mentah hingga sangat matang. Oleh karena itu, pada penelitian ini dilakukan pembuatan sistem yang dapat mendeteksi kematangan buah melinjo.

Penelitian sebelumnya [7] bertujuan untuk mendeteksi kematangan buah melinjo dengan menggunakan metode citra digital. Proses pembagian secara terstruktur mengenai kematangan menjadi 3 kelas yaitu mentah, setengah matang dan matang dimana proses tersebut dilakukan dengan menggunakan Google Collaboratory yang mengolah ruang warna RGB menjadi HSV. Pada penelitian tersebut, metode pengujian sistem yang digunakan adalah pengujian fungsional dimana pengujian dilakukan hanya dengan mengamati hasil eksekusi melalui data pengujian dan pengecekan fungsionalitas dari sistem yang dikembangkan. Tingkat akurasi yang diperoleh dari penelitian ini adalah 98,0% benar.

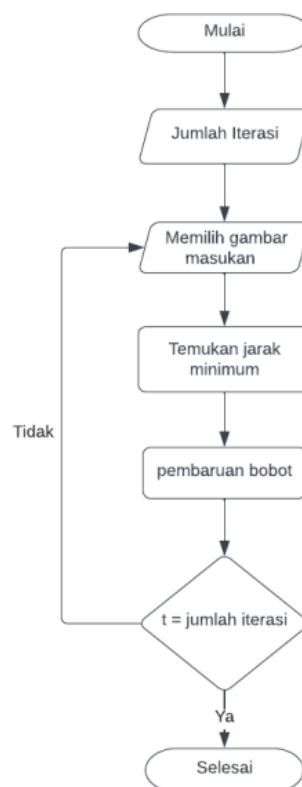
Penelitian ini mendeteksi kematangan buah melinjo menggunakan Algoritma *Self Organizing Maps* (SOM) dan ekstraksi fitur RGB (*Red, Green, Blue*) dan HSV (*Hue, Saturation, Value*). Kohonen SOM adalah salah satu metode *unsupervised network* dimana suatu proses SOM dimulai dengan melakukan pemilihan bobot node secara acak pada kohonen layer, SOM terbukti akurat dalam pengenalan pola [8-11]. Kohonen SOM diperiksa sebagai prosedur pengelompokan yang efektif. Berdasarkan peta tersebut, pengelompokan ruang faktor dari dimensi yang berbeda dilakukan, dan sampel yang representatif dibentuk [8]. Salah satu metode ekstraksi ciri adalah HSV [12-16]. Konversi citra menjadi citra HSV dipakai untuk mengekstrak objek berwarna karena lebih mudah untuk merepresentasikan warna [13]. Proses pengolahan citra dilakukan menggunakan MATLAB.

2. Metode Penelitian

Pada penelitian ini ada beberapa hal yang harus dilakukan. Tahapan pertama yaitu menyiapkan citra buah melinjo dengan beberapa sampel yang dibutuhkan sesuai dengan jenis kelas yang digunakan, kemudian dilakukan tahap *preprocessing* pada citra yaitu *resize* dan *brightening* agar citra yang digunakan memiliki citra yang bagus dan proses komputasi pada komputer dapat lebih cepat. Setelah itu, dilakukan pengelompokan citra sesuai dengan *class* yang telah disiapkan dan pemisahan data citra untuk data pelatihan dan pengujian. Pada platform MATLAB 2017a dilakukan proses pengolahan citra yaitu konversi citra dari RGB-Grayscale-Biner dan juga dilakukan operasi morfologi berupa *filling holes* dan *areaopen* agar memaksimalkan hasil segmentasi. Selanjutnya dilakukan fitur ekstraksi ciri RGB dan juga HSV dari masing-masing citra. Berikutnya yaitu membuat sistem dari Metode Algoritma *Self Organizing Map*, menjalankan proses pelatihan dan pengujian, serta akurasi pada data pelatihan dan pengujian.



Gambar 1. Diagram Alir Penelitian



Gambar 2. Diagram Alir Metode Algoritma SOM

Metode yang digunakan pada penelitian ini adalah metode Algoritma *Self Organizing Map* (SOM). Beberapa tahapan yang perlu dilakukan yaitu tahap *preprocessing*, ekstraksi fitur, dan klasifikasi. Terdapat dua proses utama dalam penelitian ini yaitu proses data pelatihan dan proses data pengujian. Proses pelatihan digunakan untuk mendapatkan nilai fitur yang dapat digunakan untuk memisahkan antara citra melinjo mentah, setengah matang, matang, dan sangat matang menggunakan data-data training. Nilai fitur tersebut kemudian digunakan dalam proses uji coba untuk mengidentifikasi kelas kematangan dari citra melinjo yang diuji.

2.1. Pengumpulan Data Citra Buah Melinjo

Pada penelitian ini ada 200 sampel data yang terbagi menjadi empat kategori, yaitu: Mentah, Setengah Matang, Matang, dan Sangat Matang. Data citra yang digunakan pada penelitian ini diambil menggunakan kamera *smartphone* Oppo f7 dengan resolusi 13 megapixel.

2.2. Data Preprocessing

Proses *preprocessing* memiliki tahapan-tahapan:

1. *Brightening* atau pengaturan tingkat kecerahan agar mutu dan kualitas citra baik
2. *Resize* atau perubahan ukuran dari 3456 x 3456 menjadi 255 x 255 agar proses komputasi lebih cepat
3. Pemilahan citra input untuk memisahkan citra yang akan digunakan sebagai sampel pelatihan dan sampel pengujian
4. Konversi citra RGB ke citra *Grayscale* untuk menyederhanakan model citra
5. Konversi citra *Grayscale* ke citra biner agar lebih sederhana lagi karena hanya ada dua nilai yaitu 0 dan 1 atau hitam dan putih
6. Melakukan operasi komplemen yaitu objek dibuat berwarna putih dan *background* berwarna hitam
7. Operasi morfologi *filling holes* untuk menghilangkan lubang yang berada pada gambar agar objek terlihat lebih utuh
8. Operasi morfologi *areaopen* untuk menghilangkan *noise*

2.3. Pembagian Fungsi Implementasi

Pada proses ini dilakukan perhitungan fitur nilai RGB menggunakan persamaan:

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \quad (1)$$

kemudian dilakukan perhitungan nilai fitur HSV menggunakan persamaan:

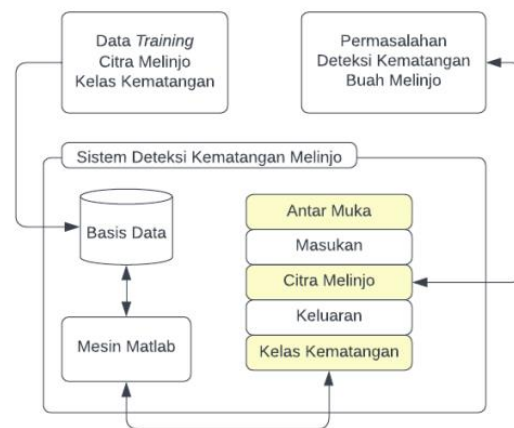
$$V = \frac{R+G+B}{3} \quad (2)$$

$$S = 1 - \frac{\min(R,G,B)}{V} \quad (3)$$

$$H = \tan\left(\frac{3(G-B)}{(R-G)+(R-B)}\right) \quad (4)$$

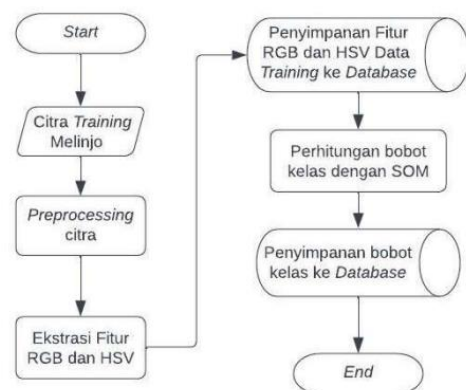
Selanjutnya, dilakukan proses klasifikasi. Dilakukan proses pelatihan SOM (*Self Organizing Map*) terhadap fitur RGB dan HSV dari seluruh data *training*. Hasil dari proses pelatihan adalah nilai fitur RGB dan HSV dari kelas mentah, setengah matang, matang, dan sangat matang. Setelah itu dilakukan proses uji coba SOM terhadap fitur RGB dan HSV dari data *testing*. Keluaran dari sistem ini adalah kelas kematangan dari data *testing*.

2.4. Keterkaitan Antar Fungsi



Gambar 3. Arsitektur Fungsi

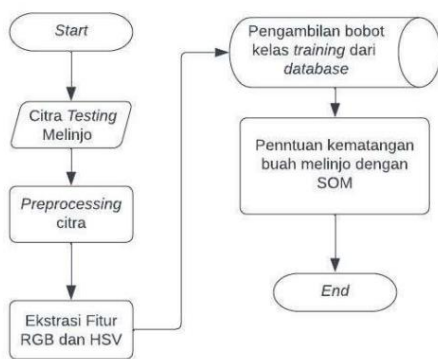
Gambar 3 menunjukkan proses arsitektur sistem dimana data berupa gambar melinjo beserta kelasnya yang digunakan sebagai data *training* dalam sistem. Pada saat melakukan *training*, pengguna memasukkan alamat *folder* yang berisi data citra melinjo yang disimpan pada basis data sistem. Bobot yang diperoleh dari proses *training* juga disimpan dalam basis data sistem untuk kemudian dipergunakan pada proses *testing* untuk melakukan deteksi kematangan buah melinjo.



Gambar 4. Diagram Alir Proses Training

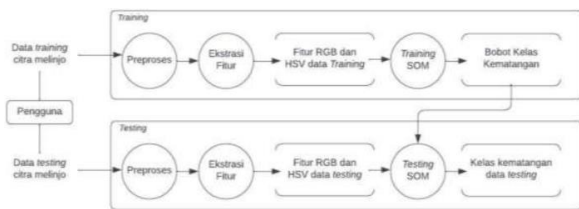
Gambar 4 menunjukkan diagram alir proses *training*. Pengguna memasukkan folder data training citra melinjo untuk digunakan dalam proses *training*. Selanjutnya data

training tersebut melalui proses *preprocessing* dan ekstraksi fitur sehingga didapatkan nilai fitur RGB data training. Nilai fitur RGB tersebut digunakan dalam proses training SOM sehingga menghasilkan data bobot kelas kematangan melinjo. Pengguna juga memasukkan sebuah citra melinjo sebagai data *testing*. Data *testing* tersebut melalui proses *preprocessing* dan ekstraksi fitur hingga menghasilkan nilai fitur RGB dari data testing tersebut. Dengan menggunakan nilai fitur RGB data testing dan data bobot kelas kematangan melinjo yang didapat sebagai keluaran dari proses *training*, dilakukan proses *testing* SOM sehingga kelas kematangan dari data testing dapat diidentifikasi.



Gambar 5. Diagram Alir Proses *Testing*

Gambar 5 menunjukkan diagram alir proses *testing*. Citra buah melinjo yang telah dimasukkan ke dalam sistem, selanjutnya citra buah melinjo yang diuji harus diambil bagian melinjo saja, maka citra buah melinjo dapat di klasifikasikan berdasarkan data sampel yang telah dimasukkan pada tahap *training*.



Gambar 6. Diagram Alir Proses *Data Flow*

3. Hasil dan Analisis

3.1. Citra Masukan

Tabel 1. Jumlah Sampel yang Digunakan

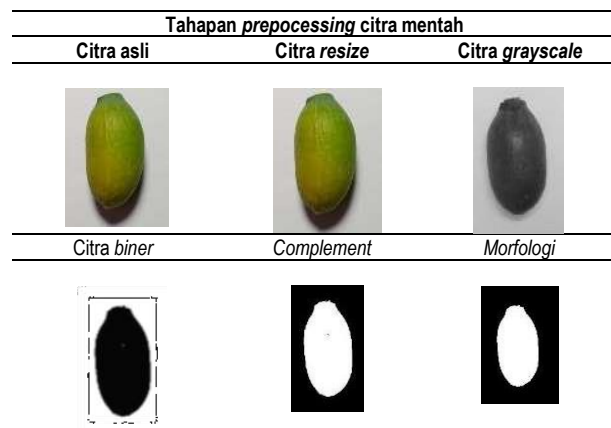
No.	Kelas	Sampel Pelatihan	Sampel Pengujian
1	Mentah	40	10
2	Setengah Matang	40	10
3	Matang	40	10
4	Sangat Matang	40	10
	Total	160	40
	Total Keseluruhan		200

Tabel 2. Citra Melinjo



3.2. Hasil *Preprocessing*

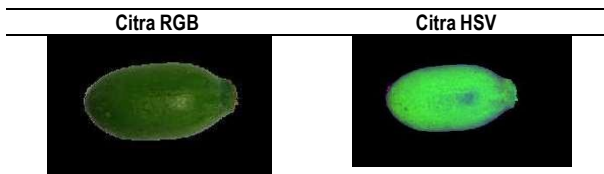
Tabel 3. Hasil *Preprocessing*



Setelah dilakukan *brightening*, pada citra yang memiliki ukuran pixels tinggi yaitu 3456 x 3456 dilakukan *resize* ukuran pixels menjadi 255 x 255 menggunakan aplikasi FastStone Photo Resizer. Ukuran citra setelah dilakukan *resize* memiliki nilai rata-rata senilai 136 kb. Untuk nilai ukuran citra tertinggi senilai 199 kb dan untuk nilai ukuran citra terendah senilai 44 kb. Setelah dilakukan tahap *resize*, selanjutnya dilakukan proses konversi fitur warna citra dari format warna RGB ke format warna keabu-abuan. *Grayscale* memiliki nilai citra warna hanya direntang 0 sampai dengan 1 sehingga disebut keabu-abuan. Selanjutnya citra *grayscale* dikonversi terhadap citra biner. Citra biner tersebut merupakan citra yang hanya bernilai 0 dan 1. Nilai gambar yang berwarna hitam memiliki nilai 0 dan putih bernilai 1. Setelah itu, dilakukan langkah komplemen atau merubah nilai citra antara background dan juga objek karena pada pengolahan citra sebuah objek diharapkan agar berwarna putih atau bernilai 1 dan background yang bernilai 0 atau hitam agar dapat memudahkan dalam pengenalan objek. Tahapan morfologi, dimana ada 2 tahap morfologi yaitu tahap *filling holes* dan *areaopen*. Tahap *filling holes* digunakan untuk menghilangkan lubang (bintik pada objek) agar menjadikan objek lebih utuh. Terakhir, dilakukan tahap *areaopen* untuk menghilangkan *noise* agar gambar lebih rapih dan detail.

3.3. Esktraksi Ciri

Tabel 4. Transformasi Citra RGB ke HSV



Pada tabel di atas merupakan citra buah melinjo mentah dari proses transformasi RGB ke HSV dengan kondisi background bernilai 0 atau hitam yang bertujuan agar proses ekstraksi ciri hanya berfokus pada citra buah saja. Pemilihan RGB dan transformasi ruang warna HSV sebagai ekstraksi ciri yang digunakan pada pengolahan citra ini. Nilai *Hue* merupakan komponen yang mengandung warna sebenarnya warna primer, warna sekunder, serta campuran warna linier. Komponen *Saturation* yang didalamnya memberikan ukuran warna putihnya atau penurunan *saturation*-nya sesuai dengan peningkatan warna putih, dan komponen *Value* merupakan komponen untuk mengukur warna kecerahannya dimana penurunan *Value* ikut sesuai dengan peningkatan kegelapan pada suatu citra. Setelah itu, membuat variabel ciri untuk setiap kelas tingkat kematangan dengan mengisi variabel tersebut dengan masing-masing ekstraksi fitur RGB dan HSV. Pada variabel ciri setiap kelas dimana baris pertama sampai keenam secara berturut-turut diisi dengan nilai ciri *Red*, *Green*, *Blue*, dan *Hue*, *Saturation*, *Value*.

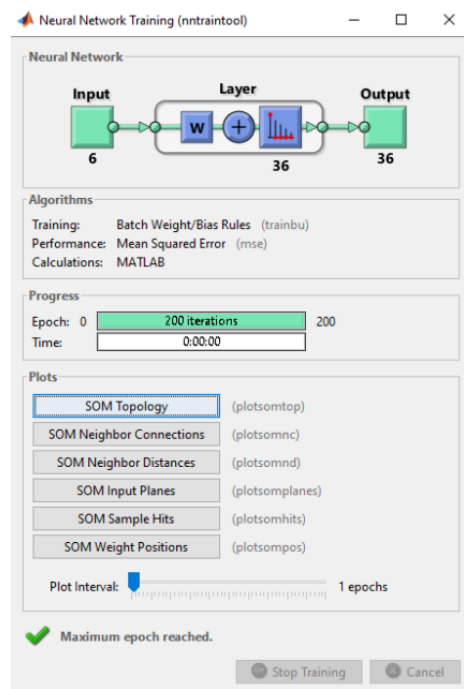
3.4. Hasil Klasifikasi Sistem

Pada proses pelatihan menggunakan citra buah melinjo yang terbagi atas 4 kelas jenis tingkat kematangan dimana setiap kelas memiliki 40 citra buah melinjo yang digunakan untuk data pelatihan. Pelatihan pada jaringan yang dibuat dalam penelitian ini menggunakan Algoritma *Self Organizing Map* dimana pada algoritma ini memiliki algoritma pembelajaran kompetitif dengan metode kohonen. SOM dapat membantu mengklasifikasikan kematangan melinjo melalui ciri warna dengan cara memetakan data ke dalam representasi yang lebih sederhana dan dimensi yang lebih rendah. Lapisan yang berisi neuron-neuron akan menyusun dirinya sendiri berdasarkan nilai input tersebut dalam *cluster* yang telah ditentukan. Selama proses penyusunan diri, *cluster* yang memiliki vektor bobot paling cocok dengan pola input (memiliki jarak yang paling dekat) akan terpilih sebagai pemenang. Neuron yang menjadi pemenang beserta neuron-neuron tetangganya akan memperbaiki bobot-bobotnya.

Tabel 5. Pengaturan Jaringan SOM

Karakteristik	Spesifikasi
<i>Dimensions</i>	[6 6]
<i>coverSteps</i>	1000(<i>default</i>)
<i>topologyFcn</i>	<i>gridtop</i>

Untuk keperluan melatih jaringan ini digunakan pengaturan perintah *train* disertai dengan *MaxEpoch* yaitu 1000(*default*), dan toleransi *error* yaitu 10^3 , *matrix default* [8 8] dan *matrix* yang digunakan [6 6] sesuai dengan ciri input ada 6 ciri yang nantinya akan membentuk *neuron output* berjumlah 36 di setiap kelasnya. Fungsi lainnya yang digunakan yaitu *topologyFcn* berupa *gridtop* dimana bentuk dari topologi yang dihasilkan pada jaringan berbentuk grid. Proses pelatihan yang telah dirancang dapat dilihat pada Gambar 7.



Gambar 7. Model Jaringan Algoritma SOM

label =	matriks =			
1	40	0	0	0
2	0	40	0	0
3	0	0	40	0
4	0	0	0	40

Gambar 8. Confusion Matrix Pelatihan

Pada pengolahan algoritma *Self Organizing Map* (SOM) dapat dilihat bahwa proses pelatihan berhenti ketika jaringan SOM ini terpenuhi. Proses *training* ini berjalan selama 8 menit 22 detik. Pada gambar di atas dapat dilihat bahwa ada 6 input yang diberikan dengan dimensi awal untuk layernya sebesar matriks [6 6] sehingga menghasilkan 36 lapisan seperti pada gambar di atas. Dikarenakan jaringan SOM ini merupakan *unsupervised learning* atau pembelajaran tak terawasi, maka proses terhenti ketika target sudah sesuai dan berhenti pada iterasi ke-200. Matlab memiliki beberapa plot dalam

menampilkan hasil akhir yaitu *SOM topologi*, *SOM neighbour connections*, *SOM form input*, *SOM sample hits*, *SOM weight position* dari *cluster* yang dibentuk.

Gambar 8 menunjukkan hasil *confusion matrix* dari proses pelatihan dengan jumlah data pelatihan sebanyak 160 citra yang terdiri dari 40 citra per kelas. Hasil setiap kelas memiliki jumlah 40 citra. Oleh karena itu, proses pelatihan bisa dikatakan berjalan dengan sangat baik karena menghasilkan nilai validasi atau akurasi 100% yang artinya seluruh citra sesuai dengan kelas yang diinginkan.

Pengujian (*testing*) pada tahap pengujian dari citra buah melinjo yang terbagi atas 4 kelas dengan per kelasnya berjumlah 8 citra data pengujian. Pada citra data pengujian dilakukan juga tahap *preprocessing* sebelum dilakukannya proses pengujian. Hasil dari tahapan proses pelatihan langsung dilakukan uji coba dengan data uji agar terlihat hasil dari proses pengujian dengan citra melinjo dengan tingkat kematangan yang berbeda dan juga pengambilan gambar dari sisi yang berbeda pula.

label =	matriks =			
1	10	0	0	0
2	0	9	1	0
3	0	0	10	0
4	0	0	0	10

Gambar 9. *Confusion Matrix* Pengujian

Gambar 9 merupakan hasil proses pengujian dengan jumlah data pelatihan sebanyak 40 citra yang terdiri dari 10 citra per kelas. Hasil setiap kelas memiliki jumlah yang berbeda dimana ada 1 citra dari 40 citra tidak sesuai dengan kelas keluaran dengan target kelas yang ditentukan. Pada kelas 2 terdapat 9 citra yang sesuai dengan target kelas dan 1 citra yang tidak sesuai dengan target kelas. Satu citra tersebut terdeteksi pada kelas keluaran 3 (kelas matang) karena citra tersebut memiliki nilai ekstraksi ciri yang mirip dalam proses pelatihan sehingga membuatnya tidak sesuai dengan target kelas yang ditentukan. Nilai validasi proses pengujian adalah:

Tabel 6. Hasil Skenario Uji Coba

Kelas	Jumlah Data Training	B		Akurasi	Jumlah Data Testing	B		Akurasi
		B	S			B	S	
1	Mentah 40	4	0	$(160/160) \times 100\% = 100\%$	Mentah 10	1	0	$(39/40) \times 100\% = 97,5\%$
2	Setengah Matang 40	4	0		Setengah Matang 10	9	1	
3	Matang 40	4	0		Matang 10	1	0	
4	Sangat Matang 40	4	0		Sangat Matang 10	1	0	

Tabel 7. Hasil Klasifikasi Sistem

Kelas	1	2	3	4	Total
1	10	0	0	0	10
2	0	9	1	0	10
3	0	0	10	0	10
4	0	0	0	10	10
Total					40

Setelah dilakukan proses pelatihan dan pengujian, hasil klasifikasi masuk pada tahap efektivitas kerja sistem pada masing-masing kelas. Efektivitas sistem dapat dilihat dari nilai *sensitivity* dengan tujuan untuk mengetahui seberapa baik *classifier* dalam mengenali data di setiap kelasnya, kemudian menghitung nilai *specificity* dengan tujuan untuk mengetahui seberapa baik *classifier* untuk tidak mengenali yang bukan pada target kelasnya, dan *accuracy* yang bertujuan untuk mengetahui tingkat kebenaran dari sistem. Untuk rumus perhitungannya mengikuti referensi [5]:

1. Kelas Mentah

$$Sensitivity = \frac{10}{10+0+0+0} \times 100\% = 100\%$$

$$Specificity = \frac{(9+1+0)+(0+10+0)+(0+0+10)}{(9+1+0)+(0+10+0)+(0+0+10)+0} \times 100\% = 100\%$$

$$Accuracy = \frac{10}{10} \times 100\% = 100\%$$

2. Kelas Setengah Matang

$$Sensitivity = \frac{9}{0+9+1+0} \times 100\% = 90\%$$

$$Specificity = \frac{(10+0+0)+(0+10+0)+(0+0+10)}{(10+0+0)+(0+10+0)+(0+0+10)+0} \times 100\% = 100\%$$

$$Accuracy = \frac{9}{10} \times 100\% = 90\%$$

3. Kelas Matang

$$Sensitivity = \frac{10}{0+0+10+0} \times 100\% = 100\%$$

$$Specificity = \frac{(10+0+0)+(0+9+0)+(0+0+10)}{(10+0+0)+(0+10+0)+(0+0+10)+1} \times 100\% = 96,67\%$$

$$Accuracy = \frac{10}{10} \times 100\% = 100\%$$

4. Kelas Sangat Matang

$$Sensitivity = \frac{10}{0+0+0+10} \times 100\% = 100\%$$

$$Specificity = \frac{(10+0+0)+(0+9+1)+(0+0+10)}{(10+0+0)+(0+9+1)+(0+0+10)+0} \times 100\% = 100\%$$

$$Accuracy = \frac{10}{10} \times 100\% = 100\%$$

Nilai rata-rata dari ketiga efektivitas sistem tersebut:

$$Sensitivity = \frac{100\%+90\%+100\%+100\%}{4} = 97,5\%$$

$$Specificity = \frac{100\%+100\%+96,67\%+100\%}{4} = 99,16\%$$

$$Accuracy = \frac{100\%+90\%+100\%+100\%}{4} = 97,5\%$$

Berdasarkan perhitungan di atas, semua komponen nilai efektivitas sistem memiliki nilai yang sangat tinggi sehingga sistem ini dapat dianggap efektif.

4. Kesimpulan

Berdasarkan hasil penelitian dapat disimpulkan bahwa perancangan untuk deteksi tingkat kematangan buah melinjo menggunakan metode algoritma *Self Organizing Map* berhasil dilakukan dengan baik menggunakan alur sesuai dengan metode penelitian yang telah dirancang. Pada penelitian yang dilakukan dengan perbandingan data *splitting* 80% untuk data *training* dan 20% untuk data *testing*. Nilai efektivitas untuk setiap kelas antara lain: kelas mentah memiliki nilai *sensitiftity* senilai 100%, *spesificity* 100 %, dan *accuracy* 100%; kelas setengah matang memiliki nilai *sensitiftity* senilai 90%, *spesificity* 100%, dan *accuracy* 90%; kelas matang memiliki nilai *sensitiftity* senilai 100%, *spesificity* 99,16%, dan *accuracy* 100%; kelas sangat matang memiliki nilai *sensitiftity* senilai 100%, *spesificity* 100%, dan *accuracy* 100%. Hasil nilai rata-rata efektivitas sistem keseluruhan sangat baik karena memiliki nilai *sensitiftity* senilai 97,5 %, *spesificity* 99,16 %, dan *accuracy* 97,5%. Hasil akurasi sistem yang dihasilkan untuk data *training* senilai 100%, sedangkan untuk akurasi data *testing* senilai 97,5%. Adapun saran yang dapat dikembangkan untuk penelitian berikutnya antara lain pada saat pengambilan citra melinjo diharapkan memiliki kualitas gambar yang lebih bagus lagi dan juga jarak pengambilan gambar yang konsisten; diharapkan pada penelitian selanjutnya dapat menggunakan fitur ciri lain selain ciri warna dan juga dapat menerapkan metode lainnya selain algoritma SOM; diharapkan pada penelitian selanjutnya dapat menggunakan *software* yang lain sehingga ada pengetahuan terbaru terkait penelitian buah melinjo ini.

Referensi

- [1]. Ni'mah, R. S., Jefri, U., Rohim, A., Wahyuni, H., & Rismayanti, N. M. Peningkatan Efektivitas dan Tingkat Produksi terhadap Penggunaan Alat Pres serta Pengembangan Produk Inovatif dari Emping Melinjo. *Batara Wisnu: Indonesian Journal of Community Services*. 2022; 2(3): 438–451.
- [2]. Siregar, Y. D. I. Pemanfaatan Ekstrak Kulit Melinjo Merah (*Gnetum gnemon*) Sebagai Pewarna Alami Pembuatan Lipstik. *Jurnal Kimia VALENSI*. 2014; 4(2): 98–108.
- [3]. Prajnaparamita, K., Susanti, S. (2021). Karakter Morfologis dan Perkembangan Anatomis Biji Melinjo (*Gnetum gnemon L.*). *Biogenesis*. 2021; 17(2): 49.
- [4]. Ika Wahyu Yuni Asri. Analisis Usaha Industri Emping Melinjo Skala Rumah Tangga di Kabupaten Magetan. Thesis. Surakarta: Universitas Sebelas Maret; 2010.
- [5]. Johan, T. M., Rifna, I. Identifikasi Kematangan Buah Tomat Berdasarkan Warna Menggunakan Metode Jaringan Syaraf Tiruan (JST) Backpropagation. *Jurnal TIKA*. 2022; 7(3), 309–315.
- [6]. Warna, F., & Neighbors, M. K. Klasifikasi Kematangan Daun Selada Berdasarkan. *Techno Xplore*. 2022; 7(1), 35–44.
- [7]. Iskandar, D., Tinggi, S., Komputer, I., Karya, C., Jakarta, E., & Author, C. Classification of Melinjo Fruits Levels Using Skin Color. *Journal of Applied Engineering and Technological Science (JAETS)*. 2022; 4(1), 123–130.
- [8]. Asri, P. P., & Wulanningrum, R. Implementasi SOM (Self Organizing Maps) untuk Identifikasi Kematangan Buah Tomat. *JTECS: Jurnal Sistem Telekomunikasi Elektronika Sistem Kontrol Power Sistem &Komputer*. 2021; 1(2), 185–192.
- [9]. Yoanta, A., & Yunitarini, R. (2017). Application of Land Grouping Using Self Organizing Map (SOM) in Sumenep Regency. *Jurnal Ilmiah Universitas Trunojoyo Madura*. 02(01), 1–5.
- [10]. Agung, A., Bagus, G., Gede, I. K., Putra, D., Dunn, I., Hasil, S. Perbandingan Metode SOM / Kohonen dengan ART 2 pada Data Mining Perusahaan Retail. *Majalah Ilmiah Teknologi Elektro*. 2017;16(02), 2–6.
- [11]. Wahyuni., Gazali, M., Hidayaturrohman, U. Pengelompokan dan Pemetaan Karakteristik Kemiskinan Di Provinsi Nusa Tenggara Barat Menggunakan Self Organizing Map (SOM) dan Biplot. *Jurnal Ilmiah Indonesia*. 2022; 7(11),2-19.
- [12]. Nurul Dzulhijjah, A., & Anraeni, S. Klasifikasi Kematangan Citra Labu Siam Menggunakan Metode KNN (K-Nearest Neighbor) dengan Ekstraksi Fitur HSV (Hue, Saturation, Value). *Buletin Sistem Informasi dan Teknologi Islam*. 2021; 2(2), 103–110.
- [13]. Prasasti, R., Wilis, N., Sains, A. Z.-S. J. Segementasi Citra Menggunakan Metode Watershed Transform dengan Kombinasi Thershold, HSV, Grayscale dan Morphology Untuk Mendeteksi Sebaran API. *Rumah Jurnal Online*. 2021; 19(1), 49–54.
- [14]. Rahmadewi, R., Sari, G. L., & Firmansyah, H. Pendeteksian Kematangan Buah Jeruk dengan Fitur Citra Kulit Buah Menggunakan Transformasi Ruang Warna HSV. *JTEV (Jurnal Teknik Elektro dan Vokasional)*. 2019; 5(1.1), 166.
- [15]. Wibowo, A., Hermanto, D. M. C., Lestari, K. I., & Wijoyo, H. (2021). Deteksi Kematangan Buah Jambu Kristal Berdasarkan Fitur Warna Menggunakan Metode Transformasi Ruang Warna HSV (Hue Saturation Value) dan K-Nearest Neighbor. *INCODING: Journal of Informatics and Computer Science Engineering*. 2021; 1(2), 76–88.
- [16]. Sanusi, H., S., S. H., & Susetianingtias, D. T. (2019). Pembuatan Aplikasi Klasifikasi Citra Daun Menggunakan Ruang Warna Rgb Dan Hsv. *Jurnal Ilmiah Informatika Komputer*. 2019; 24(3), 180–190.