

Implementasi SOM (Self Organizing Maps) untuk Identifikasi Kematangan Buah Tomat

Puput Puji Asri¹, Resty Wulanningrum²

^{1,2}Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri

E-mail: *¹puput.jias@gmail.com, ²resty0601@gmail.com

Abstrak – Tomat salah satu buah sekaligus sayuran dengan harga yang ekonomis dan sering kali kita jumpai. Harganya yang tidak terlalu mahal dan banyak mengandung vitamin dan mineral serta antioksidan yang tinggi, membuat tomat banyak dibeli dimasyarakat. Sehingga membuat petani buah tomat untuk terus meningkatkan mutu dan kualitas terhadap pelayanan kepada kosumen. Tetapi kadang banyak petani yang menjual buah tomatnya dengan kematangan yang tidak seragam, sehingga kadang yang sudah matang tertindih oleh yang mengkal atau mentah, sehingga banyak yang membusuk. Dari permasalahan ini, maka penulis ingin membuat sebuah aplikasi untuk memilah kematangan buah tomat mentah, mengkal dan matang. Penggunaan metode *Self Organizing Map* (SOM) ini bertujuan untuk melakukan pengelompokkan antara tomat yang mentah, mengkal dan matang. SOM tidak akan menghentikan proses iterasinya selama jumlah iterasinya belum mencapai target yang diharapkan. Hasil penelitian yang dihasilkan adalah sebuah aplikasi identifikasi kematangan buah tomat untuk mempermudah petani atau masyarakat luas dengan cepat. Implementasi pengujian identifikasi kematangan buah tomat ini adalah 91,11%.

Kata Kunci — buah tomat, euclidean distance, SOM

Abstract – Tomato is one of the fruits and vegetables at an economical price and we often encounter. The price is not too expensive and contains lots of vitamins and minerals as well as high antioxidants, making tomatoes widely purchased in the community. So that it makes tomato fruit farmers to continue to improve the quality and quality of service to consumers. But sometimes many farmers sell their tomatoes with non-uniform maturity, so that sometimes the ripe ones are crushed by the ripe or raw ones, so that many rot. From this problem, the author wants to create an application to sort the ripeness of raw, ripe and ripe tomatoes. The use of the *Self Organizing Map* (SOM) method aims to classify tomatoes that are raw, ripe and ripe. SOM will not stop the iteration process as long as the number of iterations has not reached the expected target. The result of the research is an application to identify the ripeness of tomatoes to make it easier for farmers or the wider community quickly. The implementation of the tomato fruit maturity identification test was 91.11%.

Keywords — tomato fruit, euclidean distance, SOM

1. PENDAHULUAN

Perkembangan arus teknologi informasi dan komunikasi pada era globalisasi [1] saat ini sudah sangat meningkat dalam hal pengetahuan dan teknologi yang berkaitan dengan pengambilan, pengumpulan, pengolahan penyimpanan dan penyebaran informasi. Adanya teknologi yang berkembang pesat, akan mempermudah masyarakat dalam mendapatkan informasi dan berita di internet. Kemajuan teknologi yang seperti ini memberikan dampak yang sangat baik juga teradap kemajuan kehidupan masyarakat. Begitu pula dalam hal bidang pertanian. Seperti halnya buah tomat, merupakan salah satu jenis buah yang memiliki harga ekonomis, sehingga paling banyak digemari oleh semua orang baik untuk dimakan langsung, dimasak atau dijual.

Tomat atau *Solanum Lycopersicum* syn. *Lycopersicum esculentum* merupakan tumbuhan dari keluarga *solanaceae*. Tomat merupakan salah satu tumbuhan asli Amerika Selatan dan Tengah serta Peru hingga Meksiko. Tanaman ini memiliki siklus hidup yang singkat dan mampu tumbuh dengan ketinggian 1 hingga 3 meter [2]. Tomat merupakan tanaman sayuran buah yang sangat dibutuhkan [3] oleh tubuh manusia, karena buah tomat mempunyai kandungan berbagai gizi maupun nutrisi untuk memenuhi kebutuhan tubuh manusia demi kelangsungan hidup yang sehat. Dengan mengkonsumsi tomat secara rutin setiap hari dapat menjaga kesehatan tubuh manusia seperti menjaga kesehatan mata, kulit, jantung menurunkan kadar kolesterol, dan memperlancar sistem pencernaan serta memperkuat tulang [4].

Setiap petani buah pasti berusaha untuk meningkatkan kualitas hasil pertaniannya baik dari segi kualitas buah itu sendiri maupun kualitas pelayanan terhadap konsumen. Akan tetapi dengan adanya keterbatasan kemampuan yang dimiliki oleh manusia serta banyaknya jumlah buah tomat yang akan dipasarkan kepada konsumen, petani sering menghadapi permasalahan dalam melakukan pemilahan buah tomat yang telah matang, mengkal, dan mentah. Karena antara tomat matang, mengkal, dan mentah tersebut mempunyai kematangan, kualitas dan harga yang berbeda-beda sehingga masing-masing dari buah tomat tersebut harus dipisahkan. Untuk mengatasi permasalahan tersebut dibutuhkan suatu metode yang mampu melakukan pengelompokan secara akurat antara buah tomat matang, mengkal, dan mentah.

Dalam penelitian ini penulis melakukan identifikasi tingkat kematangan pada buah tomat berdasarkan pada warna kulit buah tomat, karena warna kulit tersebut menjadi salah satu ciri yang mudah untuk diketahui dalam menentukan apakah buah siap dipasarkan atau belum. Sehingga dari sini penulis menerapkan metode pengelompokan (*clustering*), yaitu dengan menggunakan metode *Self Organizing Maps* (SOM) untuk melakukan pengelompokan antara buah tomat yang mentah, mengkal, dan matang. Berdasarkan masalah diatas penulis berusaha membuat *software* tentang deteksi kematangan buah tomat untuk menentukan apakah buah tomat siap dijual atau dipasarkan ke konsumen atau belum berdasarkan warna dengan menggunakan *Self Organizing Maps* (SOM).

2. METODE PENELITIAN

Metode dalam penelitian ini meliputi beberapa bagian, yaitu.

2.1. *Self Organizing Maps* (SOM)

Kohonen SOM adalah salah satu metode *unsupervised network* dimana suatu proses *self organizing* dimulai dengan melakukan pemilihan bobot *node* secara acak pada kohonen *layer*, SOM terbukti akurat dalam pengenalan pola [5], [6], [7] namun kohonen SOM mempunyai kelemahan perlunya data pelatihan yang besar dan waktu komputasi yang tinggi [8]. Setiap neuron dalam SOM akan mewakili satu kelompok. Dalam SOM ada K neuron yang tersusun dalam larik satu dimensi atau dua dimensi. Sinyal masukan untuk setiap neuron dilewatkan pada bobot dengan jumlah elemen bobot yang sama dengan N fitur (Kohonen, 1989) sehingga arsitektur SOM akan membutuhkan $N \times K$ bobot. Nilai data masukan yang dilewatkan melalui bobot-bobot yang berkorelasi dengan neuron akan menghasilkan nilai keluaran dari setiap neuron, Neuron dengan nilai terkecil dianggap sebagai neuron pemenang yang selanjutnya akan melakukan pembaharuan bobotnya pada iterasi tersebut, bersama dengan tetangga-tetangga yang didefinisikan dalam arsitekturnya. Arsitektur SOM dapat berupa larik satu atau dua dimensi [9].

2.2. *Perencanaan dan Tahap Implementasi*

Terdapat beberapa tahapan dalam proses identifikasi kematangan buah tomat dengan menerapkan metode SOM (*Self Organizing Maps*), yaitu tahap preproses, ekstraksi fitur, dan klasifikasi.



Gambar 1. Tahapan Sistem

Tahapan-tahapan tersebut ditunjukkan pada Gambar 1 di atas. Masukkan dari sistem ini adalah citra buah tomat dan keluarannya adalah kelas kematangan dari citra buah tomat tersebut.

2.2.1. Pembagian Fungsi Implementasi

Pada sistem identifikasi kematangan buah tomat, terdapat dua proses utama antara lain proses pelatihan (*training*) dan proses uji coba (*testing*). Tahapan yang dimiliki oleh kedua proses tersebut sama, yaitu yang ditunjukkan pada Gambar 1. Proses pelatihan digunakan untuk mendapatkan nilai fitur yang dapat digunakan untuk memisahkan antara tomat mentah, mangkal, dan matang menggunakan data-data *training*. Nilai fitur tersebut kemudian digunakan dalam proses uji coba untuk mengidentifikasi kelas kematangan dari citra buah tomat yang diuji.

Pada tahap pertama yaitu preproses, dilakukan pemilihan citra masukan. Citra masukan tersebut kemudian dibagi menjadi 3 buah yaitu citra merah, hijau, dan biru karena input citra tomat yang digunakan dalam sistem ini merupakan citra berwarna. Selanjutnya, pada tahap ekstraksi fitur maka jumlah nilai piksel dari masing-masing citra merah, hijau, dan biru dihitung untuk memperoleh nilai dari fitur RGB (*red, green, blue*). Digunakan Persamaan 1-3 untuk menghitung nilai fitur *red, green, dan blue* secara berturut-turut [10] dimana R adalah nilai fitur *red*, G adalah nilai fitur *green*, B adalah nilai fitur *blue*, *red* merupakan jumlah nilai piksel citra merah, *green* merupakan jumlah nilai piksel citra hijau, dan *blue* merupakan jumlah nilai piksel citra biru.

$$R = \frac{red}{red+green+blue} \dots\dots\dots (1)$$

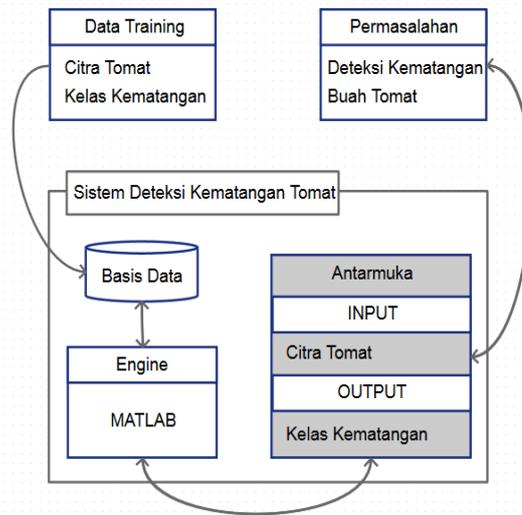
$$G = \frac{green}{red+green+blue} \dots\dots\dots (2)$$

$$B = \frac{blue}{red+green+blue} \dots\dots\dots (3)$$

Setelah didapatkan fitur RGB, maka dilakukan proses klasifikasi. Dilakukan proses pelatihan SOM (*Self Organizing Maps*) terhadap fitur RGB dari seluruh data *training*. Hasil dari proses pelatihan adalah nilai fitur RGB dari kelas mentah, mangkal, dan matang. Setelah itu dilakukan proses uji coba SOM terhadap fitur RGB dari data *testing*. *Output* dari sistem ini adalah kelas kematangan dari data *testing*.

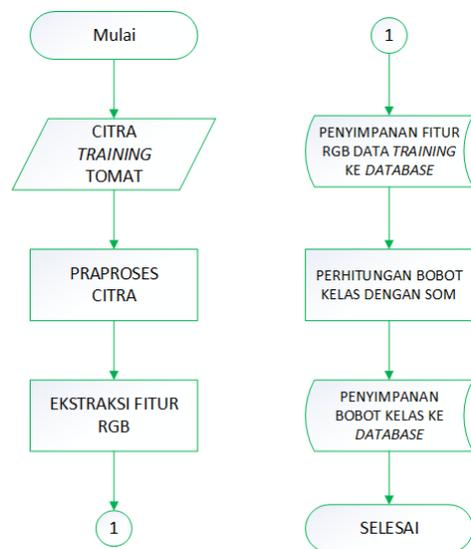
2.2.2. Keterkaitan Antar Fungsi

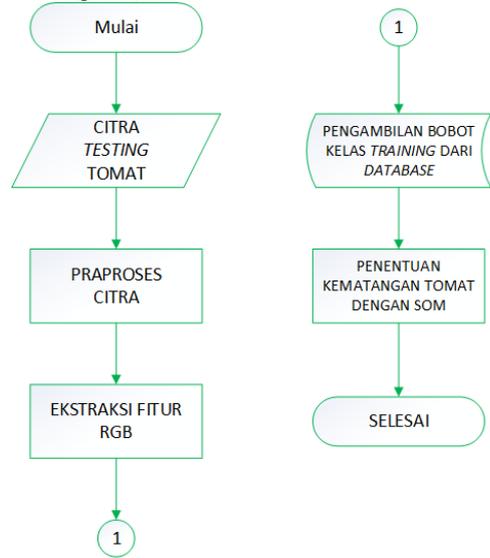
Gambar 2 menunjukkan arsitektur dari sistem identifikasi kematangan buah tomat. Arsitektur dalam sistem ini memiliki 3 komponen, yaitu basis data, *engine*, dan antarmuka pengguna. Arsitektur sistem ditunjukkan pada Gambar 1. Data berupa gambar tomat beserta kelasnya digunakan sebagai data *training* dalam sistem. Pada saat melakukan *training*, pengguna memasukkan alamat *folder* yang berisi data citra tomat yang kemudian akan disimpan kedalam basis data sistem. Bobot yang diperoleh dari proses *training* juga akan disimpan dalam basis data sistem untuk kemudian dipergunakan pada proses *testing* untuk melakukan deteksi kematangan buah tomat. *Engine* yang digunakan dalam sistem ini adalah *software Matlab*.



Gambar 2. Arsitektur sistem

Secara umum terdapat dua buah proses utama dalam sistem deteksi kematangan buah tomat, yaitu proses *training* dengan alur metode yang ditunjukkan oleh Gambar 3 dan proses *testing* dengan alur metode yang ditunjukkan pada Gambar 4. Baik proses *training* maupun testing sama-sama melalui tahap preproses dan ekstraksi fitur sebelum memasuki tahap klasifikasi. Keterkaitan antara proses *training* dan *testing* secara lebih jelas dapat dilihat pada *data flow diagram* (DFD) *level 1* yang ditunjukkan oleh Gambar 5.

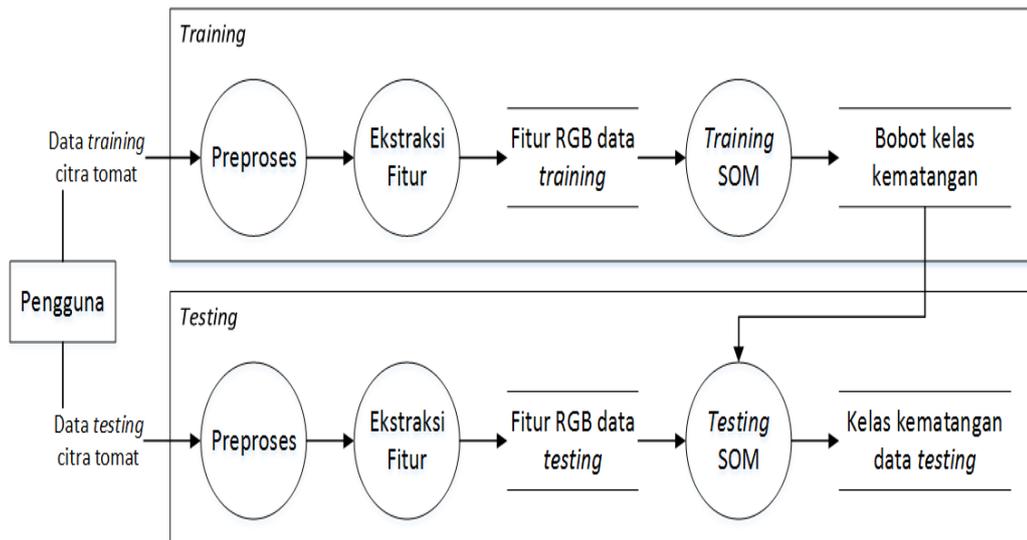
Gambar 3. Diagram alir proses *training*



Gambar 4. Diagram alir proses *testing*

Pengguna memasukkan *folder* data *training* citra tomat untuk digunakan dalam proses *training*. Selanjutnya data *training* tersebut akan melalui proses preproses 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 tomat.

Pengguna juga memasukkan sebuah citra tomat sebagai data *testing*. Data *testing* tersebut akan melalui proses preproses dan ekstraksi fitur hingga menghasilkan nilai fitur RGB dari data *testing* tersebut. Menggunakan nilai fitur RGB data *testing* dan data bobot kelas kematangan tomat yang didapat sebagai *output* dari proses *training*, akan dilakukan proses *testing* SOM sehingga kelas kematangan dari data *testing* dapat diidentifikasi.



Gambar 5. DFD Level 1 Sistem Identifikasi Kematangan Buah Tomat Menggunakan SOM

2.3. Skenario Uji Coba

Pada tahapan ini akan dijelaskan mengenai proses *testing* serta evaluasi hasil skenario uji coba perangkat lunak. Pembahasan akan meliputi data uji coba, dan pemaparan hasil uji coba.

Skenario uji coba ini memberikan penjelasan tentang proses uji gambar *testing* untuk mengetahui tingkat akurasi dari aplikasi yang telah dibuat. Untuk penjelasan lebih lanjut dituangkan dalam tabel berikut.

Tabel 1. Skenario ujicoba

No	Skenario	Training	Jml	Testing	Jml
1	1	Tomat Matang 1_1, 1_2, 1_3, 1_4, 1_5, 1_6, 1_7, 1_8, 1_9, 1_10, 1_11, 1_12, 1_13, 1_14, 1_15, 1_16, 1_17, 1_18, 1_19, 1_20 Tomat Mengkal 2_1, 2_2, 2_3, 2_4, 2_5, 2_6, 2_7, 2_8, 2_9, 2_10, 2_11, 2_12, 2_13, 2_14, 2_15, 2_16, 2_17, 2_18, 2_19, 2_20 Tomat Mentah 3_1, 3_2, 3_3, 3_4, 3_5, 3_6, 3_7, 3_8, 3_9, 3_10, 3_11, 3_12, 3_13, 3_14, 3_15, 3_16, 3_17, 3_18, 3_19, 3_20	60	Tomat Matang 1_21, 1_22, 1_23, 1_24, 1_25 Tomat Mengkal 2_21, 2_22, 2_23, 2_24, 2_25 Tomat Mentah 3_21, 3_22, 3_23, 3_24, 3_25,	15

Tabel 1 diatas merupakan tabel skenario uji coba yang dilakukan dengan total gambar keseluruhan sebanyak 75 gambar. Dengan pembagian sebagai berikut :

- Skenario 1
60 gambar *training* dan 15 gambar *testing*.
- Skenario 2
45 gambar *training* dan 30 gambar *testing*.
- Skenario 3
30 gambar *training* dan 45 gambar *testing*.

3. HASIL DAN PEMBAHASAN

Implementasi program terbagi menjadi 2, yaitu tampilan *training* dan tampilan *testing*.

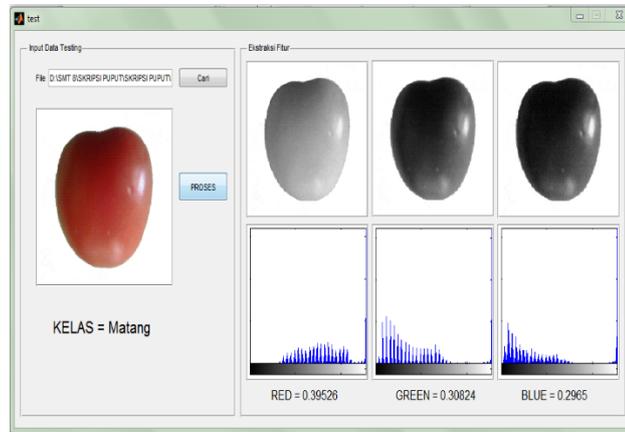
The screenshot shows a window titled 'train' with two main sections. The top section, 'Input Data Training', contains a 'Folder' text box with the path 'D:\SMT 8\SKRIPSI PUPUT\SKRIPSI PUPUT\Program\Train' and a 'Cari' button. Below this is a large blue 'PROSES' button. The bottom section, 'Bobot Kelas', contains a table with three columns: 'RED', 'GREEN', and 'BLUE'. The rows represent 'Kelas Matang', 'Kelas Mengkal', and 'Kelas Mentah'.

	RED	GREEN	BLUE
Kelas Matang	0.63354	0.22307	0.14339
Kelas Mengkal	0.55512	0.36384	0.081041
Kelas Mentah	0.4081	0.47015	0.12175

Gambar 6. Form menu training

Pada Gambar 6 di atas menyajikan tampilan dari menu *training* yang dipakai untuk melakukan proses pembelajaran pada data *training*. Pengguna diminta untuk memasukkan alamat *folder* yang berisi citra tomat untuk data *training* dengan menekan tombol "Cari", kemudian menekan tombol "Proses" untuk menentukan bobot dari kelas kematangan tomat. Bobot ini

kemudian disimpan kedalam basis data sistem dan kemudian akan digunakan saat melakukan proses uji coba. Untuk hasilnya setelah di proses dapat dilihat pada Gambar 7.



Gambar 7. Form menu testing

Pada Gambar 7 di atas adalah tampilan dari menu *testing* yang dipergunakan untuk melakukan uji coba kematangan buah tomat. Pengguna diminta untuk memasukkan alamat dari *file* gambar buah tomat yang hendak diuji, dan kemudian akan tampil gambar dari buah tomat yang hendak diuji tersebut. Setelah pengguna menekan tombol “Proses”, maka akan dilakukan proses klasifikasi kematangan buah tomat. Pada kolom sebelah kanan akan ditampilkan citra RGB hasil *grayscale* dan nilai dari masing-masing fitur RGB beserta histogramnya. Dan yang terakhir akan muncul kelas kematangan dari buah tomat yang sedang diuji coba. Untuk hasilnya setelah di proses dapat dilihat pada Gambar 7.

Proses uji coba sistem identifikasi kematangan tomat ini menggunakan 75 data berupa gambar dari buah tomat. Pembagian pertama untuk gambar *training* sebanyak 60 gambar dan untuk gambar *testing* sebanyak 15, pembagian kedua untuk gambar *training* sebanyak 45 gambar dan untuk gambar *testing* sebanyak 30, dan pembagian ketiga untuk gambar *training* sebanyak 30 gambar dan untuk gambar *testing* sebanyak 45 buah tomat.

Seluruh gambar yang digunakan dalam sistem ini baik gambar *training* maupun gambar *testing* merupakan gambar yang diambil dari beberapa kematangan yang berbeda dengan bentuk yang berbeda pula. Dalam proses uji coba ini sample gambar diambil dari buah tomat dengan kematangan yang berbeda dengan bentuk yang berbeda pula.

Tabel 2. Hasil Skenario Ujicoba

No	Σ Data Training	Σ Data Testing	B	S	Akurasi
1	Matang 20	Matang 5	5	0	$(11/15)*100\%$ = 73.33%
	Mengkal 20	Mengkal 5	1	4	
	Mentah 20	Mentah 5	5	0	
2	Matang 15	Matang 10	10	0	$(26/30)*100\%$ = 86.67%
	Mengkal 15	Mengkal 10	6	4	
	Mentah 15	Mentah 10	10	0	
3	Matang 10	Matang 15	15	0	$(41/45)*100\%$ = 91.11%
	Mengkal 10	Mengkal 15	11	4	
	Mentah 10	Mentah 15	15	0	

Dari skenario uji coba 1 yang telah dilakukan terhadap 15 data *testing*, data yang dapat diidentifikasi kematangannya oleh sistem berjumlah 11, sedangkan pada skenario uji coba 2 yang telah dilakukan pada 30 data *testing*, data yang dapat diidentifikasi kematangannya oleh sistem berjumlah 26, dan pada skenario uji coba 3 yang telah dilakukan pada 45 data *testing*, data yang dapat diidentifikasi kematangannya oleh sistem berjumlah 41. Untuk menambah tingkat akurasi aplikasi, *user* dapat menambah jumlah gambar *training*. Jadi kesimpulannya, pembagian data *training* dan data *testing* berpengaruh terhadap hasil akurasi dari sistem.

4. KESIMPULAN

Dari hasil perancangan sistem identifikasi kematangan buah tomat ini dihasilkan sebuah aplikasi yang dapat mengidentifikasi kematangan buah tomat sesuai kelasnya, yaitu kelas matang, mengkal, dan mentah. Aplikasi ini juga mempermudah pengguna atau *user* dalam memilah-milah buah tomat sesuai warnanya dengan cepat, tepat, dan efisien sehingga mempercepat proses pemasaran ke konsumen. Dari hasil pengujian yang dilakukan pada 3 skenario didapatkan tingkat keakuratan sebesar 73.33% dengan data *training* sebanyak 60 dan data *testing* sebanyak 15, tingkat keakuratan sebesar 86.67% dengan data *training* sebanyak 45 dan data *testing* sebanyak 30, dan tingkat keakuratan sebesar 91,11% dengan data *training* sebanyak 30 dan data *testing* sebanyak 45 buah tomat. Pembagian jumlah data *training* dan data *testing* dapat berpengaruh terhadap hasil akurasi, jarak pengambilan gambar berpengaruh terhadap hasil, dan pengaruh cahaya sangat rentan terhadap hasil pengujian.

DAFTAR PUSTAKA

- [1] Setyanto E, N Rasyidah, M Sulhan. 2017. Aplikasi TIK dalam Manajemen Pendidikan Dasar dan Menengah. HIKMAH: Jurnal Pendidikan Islam Vol. 6, No.2. STAI TUANKU TAMBUSAI. RIAU.
 - [2] Mappiratu, Nurhaeni, and I. Israwaty, "Pemanfaatan tomat afkiran untuk produksi likopen," Media Litbang Sulteng, vol. 3, no. 1, pp. 64–69, 2010.
 - [3] Smith, Andrew F. 1994. *The Tomato in America. Amerika: University of Illinois Press*. Online, <https://id.wikipedia.org/wiki/Tomat>
 - [4] Anonim. 2013. Pedoman Teknis Budidaya Tomat. [https:// tabloitsinartani.com/detail/indeks/kebun/31-budidaya-tomat/](https://tabloitsinartani.com/detail/indeks/kebun/31-budidaya-tomat/)
 - [5] Kim, K. I., Jung K., & Kim, H. J., 2002, *Face Recognition Using Kernel Principal Component Analysis, IEEE Signal Processing letters*, Vol 9, No. 2.
 - [6] Majumder, A., Behera, L., & Subramanian, V. K., 2014. *Local Binary Pattern based Facial Expression Recognition using Self-Organizing Map. International Joint Conference on Neural Networks (IJCNN)*. 2375- 2382
 - [7] Chelali, F. Z., Djeradi, A., & Cherabit, N., 2015, *Investigation of DCT/PCA combined with Kohonen classifier for human identification*. 978-1-4673-6673-1.
 - [8] Harahap, M., Tulus, Budiarti, E., 2014, Analisa Perbandingan Pengenalan Tulisan Tangan dengan Kohonen SOM dan MDF, Tesis Megister Teknik Informatika Universitas Sumatera Utara.
 - [9] Prasetyo, Eko. 2012. DATA MINING Konsep dan Aplikasi Menggunakan MATLAB. Yogyakarta: ANDI Yogyakarta.
 - [10] D. Erwanto, S. A. D. Prasetyowati, dan E. N. Budi Susila, "Utilization of Digital Image Processing in Process of Quality Control of the Primary Packaging of Drug Using Color Normalization Method," in *IOP Conference Series: Materials Science and Engineering*, 2017.
-