

Menghitung Jumlah Buah Cabe Berwarna Hijau Menggunakan Metode Transformasi Ruang Warna RGB

Atika Mardhotillah¹, Anditya Atfianto², Kurniawan Nur Rahmadhani³

^{1,2,3}Fakultas Informatika, Universitas Telkom, Bandung

¹amardhotillah5@gmail.com, ²anditya@telkomuniversity.ac.id, ³andiess2006@gmail.com

Abstrak

Perhitungan otomatis hasil panen dan tanaman seperti buah dan sayur merupakan sistem yang dapat membantu pekerjaan untuk mendapatkan informasi. Transformasi ruang warna merupakan pengolahan citra membantu dalam mendeteksi warna dalam citra dan mengolahnya sehingga mempermudah proses identifikasi suatu objek dalam citra tersebut. Hasil penelitian menunjukkan bahwa metode ini dapat diterapkan pada perhitungan jumlah buah cabe hijau dengan akurasi 87,9%.

Kata Kunci, hitung jumlah buah cabe, cabe hijau, transformasi ruang warna.

Abstract

Automatic calculation of crops and crops such as fruits and vegetables is a system that can help the job to obtain information. Transformation of color space is an image processing helps in detecting the color in the image and process it so as to facilitate the process of identifying an object in the image. The results showed that this method can be applied on the calculation of the number of green chili with an accuracy of 87.9%.

Keywords, counting green chillies, green chilli, color space transformation.

1. Pendahuluan

Cara menghitung buah yang berbentuk tidak bulat dilakukan dengan beberapa cara, baik secara manual maupun secara otomatis. Cara manual dilakukan berdasarkan pengamatan visual secara langsung pada pohon buah yang akan dihitung. Penghitungan buah di sektor pertanian dengan cara manual membutuhkan waktu yang cukup lama. Perkembangan teknologi informasi memungkinkan menghitung buah secara otomatis berdasarkan klasifikasi pola bentuk buah dengan bantuan komputer.

Beberapa metode klasifikasi pola dengan identifikasi warna akhir-akhir ini telah berkembang sangat pesat. Salah satu perkembangan metode klasifikasi warna adalah metode transformasi ruang warna rgb yaitu merupakan metode pemisahan warna red, blue, green untuk menentukan objek yang akan dideteksi, beberapa proses preprocessing awal seperti: segmentasi citra, filtering dan akuisisi citra dilakukan untuk memperjelas objek yang akan di hitung..

Pada penelitian ini yang menjadi objek untuk diamati adalah buah cabe berwarna hijau yang berjenis rawit, karena buah cabe berwarna hijau berjenis rawit memiliki bentuk yang tidak bulat dan tingkat kemiripan buah cabe berwarna hijau dengan *background* yaitu batang dan daun adalah 70-90%. Informasi gambar yang dipergunakan pada penelitian ini diperoleh dengan menggunakan kamera, teknik pengambilan gambar menyesuaikan dengan kontras cahaya.

1.2 Latar Belakang

Pemetaan hasil panen buah yang dilakukan dengan cara mengidentifikasi jumlah buah/sayuran di pohon/tanaman merupakan salah satu cara yang dapat dilakukan, untuk memprediksi hasil sebelum panen. Dengan prediksi tersebut, petani dapat mengatur penggunaan pupuk dan sumber daya lainnya, sehingga petani dapat mengelola pertanian dengan baik dan lebih efisien untuk meningkatkan hasil dan keuntungan [1]. Saat ini perkembangan teknologi semakin meningkat dan dapat dimanfaatkan untuk mempermudah melakukan pekerjaan disegala bidang termasuk di bidang pertanian seperti teknologi robotika dan sistem estimasi lainnya.

Sistem estimasi dapat dipergunakan untuk memprediksi jumlah buah berbentuk bulat maupun tidak bulat yang masih berada pada pohon/batang. Untuk menghitung atau memprediksi jumlah buah cabe hijau pada pohon/batang memiliki kerumitan yang lebih tinggi di banding buah lainnya. Sistem estimasi yang dapat dipergunakan dalam menghitung atau memprediksi jumlah buah yang berbentuk tidak bulat dan berwarna hijau pada pohon/batang adalah menggunakan metode Transformasi Ruang Warna RGB.

Adapun objek yang digunakan dalam memprediksi/menghitung jumlah buah tidak bulat dan berwarna hijau di pohon/batang pada penelitian ini adalah buah cabe hijau berjenis rawit.

1.3 Topik dan Batasannya

Penggunaan sistem estimasi dalam memprediksi atau menghitung jumlah buah berbentuk bulat pada pohon /batang telah dilakukan dengan berbagai macam cara dan metode, Sedangkan untuk menghitung/memprediksi jumlah buah berbentuk tidak bulat berwarna hijau pada pohon/batang memberikan peluang untuk dilakukan penelitian, sehingga topik penelitian ini adalah menghitung/memprediksi buah berbentuk tidak bulat berwarna hijau menggunakan metode Transformasi Ruang Warna RGB.

Objek buah yang digunakan adalah buah cabe hijau berjenis rawit, dikarenakan buah cabe berjenis rawit merupakan buah yang berbentuk tidak bulat dan memiliki warna hijau buah dan latar belakang (daun dan batang) yang hampir serupa 80 – 90% kemiripan warna hijau.

Penelitian ini tidak mengatasi buah yang > 80% tertutup oleh batang dan daun dan gambar cabe yang kabur atau bias dikatakan *blur*. Buah cabe yang tertutup daun dan batang hingga > 80% akan menyulitkan proses klasifikasi untuk mendeteksi buah berbentuk tidak bulat berwarna hijau tersebut.

1.4 Tujuan

Penelitian ini bertujuan untuk mengimplementasikan perhitungan jumlah Buah Cabe Berwarna Hijau Menggunakan Metode Transformasi Ruang Warna RGB.

1.5 Organisasi Tulisan

Pada bagian 2 akan dijelaskan penelitian sebelumnya serta landasan teori yang terkait dengan penelitian ini. Pada bagian 3 dijelaskan proses sistem yang dibangun, pada bagian 4 dijelaskan mengenai observasi sistem dan pengujian performansi. Terakhir pada bagian 5 terdapat kesimpulan yang menjawab permasalahan pada penelitian ini serta saran dari penulis untuk penelitian terkait selanjutnya.

2. Studi Terkait

2.1 Citra

Citra digital merupakan representatif dari citra yang diambil oleh mesin dengan bentuk pendekatan berdasarkan sampling dan kuantisasi. Sampling menyatakan besarnya kotak-kotak yang disusun dalam baris dan kolom. Dengan kata lain, sampling pada citra menyatakan besar kecilnya ukuran *pixel* (titik) pada citra, dan kuantisasi menyatakan besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan (*grayscale*) sesuai dengan jumlah bit biner yang digunakan oleh mesin, dengan kata lain kuantisasi pada citra menyatakan jumlah warna yang ada pada citra [1].

Citra Warna (RGB) adalah sebuah citra yang setiap pikselnya merupakan gabungan dari tiga komponen warna yaitu merah (red), hijau (green), biru (blue). Masing-masing komponen memiliki rentang warna antara 0-255 untuk mempresentasikan tingkatan warna masing-masingnya [4].

Citra abu adalah sebuah citra yang menangani gradiasi warna hitam dan putih, yang menghasilkan efek warna abu-abu yang warnanya dinyatakan dengan intensitas. Intensitas citra berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih [3].

Citra biner adalah citra dengan setiap piksel hanya dinyatakan dengan sebuah nilai dari dua kemungkinan (yaitu nilai 0 dan 1). Nilai 0 menyatakan warna hitam dan nilai 1 menyatakan warna putih. Citra jenis ini banyak dipakai dalam pemrosesan citra, misalnya untuk kepentingan memperoleh tepi bentuk suatu objek [3].

2.2 Segmentasi Warna RGB

Segmentasi warna, ada bermacam-macam model warna. Model RGB (Red Green Blue) merupakan model yang banyak digunakan, salah satunya adalah monitor [1]. Pada model ini untuk merepresentasikan gambar menggunakan 3 buah komponen warna tersebut. Selain model RGB terdapat juga model normalisasi RGB dimana model ini terdapat 3 komponen yaitu, r, g, b yang merepresentasikan prosentase dari sebuah piksel pada citra digital [5][8]. Nilai-nilai tersebut mengikuti persamaan-persamaan dibawah ini :

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

Sehingga : $r + g + b = 1$

Dengan demikian berdasarkan persamaan 7 maka cukup hanya menggunakan r dan g saja, karena nilai b bisa didapatkan dengan menggunakan $b = 1 - r$

2.3 Ruang Warna HSV

HSV adalah singkatan dari warna (hue), saturasi (saturation), dan nilai (value), hsv sering disebut HSB yaitu B untuk brightness [1]. Model warna HSV mendefinisikan warna dalam terminology Hue, Saturation dan Value. Hue menyatakan warna sebenarnya, seperti merah, violet, dan kuning. Hue digunakan untuk

membedakan warna-warna dan menentukan kemerahan (redness), kehijauan (greenness), dsb, dari cahaya. Hue berasosiasi dengan panjang gelombang cahaya. Saturation menyatakan tingkat kemurnian suatu warna, yaitu mengindikasikan seberapa banyak warna putih diberikan pada warna. Value adalah atribut yang menyatakan banyaknya cahaya yang diterima oleh mata tanpa memperdulikan warna.

2.4 Dataset

Tanaman yang digunakan dalam penelitian ini adalah tanaman cabe, buah cabe yang berjenis rawit. Buah cabe berjenis rawit berwarna hijau yang tingkat kemiripan buah dengan *background* (daun dan batang) 70-90%, selain itu juga buah cabe berbentuk tidak bulat. Pada penelitian buah cabe yang digunakan, yaitu rawit yang berwarna hijau tua seperti daun dan batang.

2.5 Perhitungan Performasi

Perhitungan performasi merupakan perhitungan seberapa tepat dan bagusnya sistem dalam melakukan tujuan pembuatannya [2]. Untuk perhitungan performasi pada penelitian tugas akhir ini menggunakan *confussion matrix* yaitu sebuah tabel yang digunakan untuk menghitung performasi sistem. Berikut rumus perhitungan performasi dari *confussion matrix*.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (2.1)$$

Table 2.1 Confusion Matrix

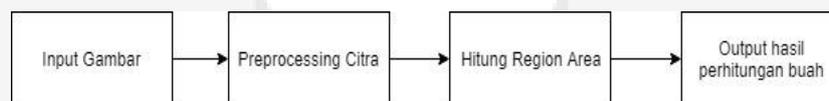
		Predicted	
		True	False
Actual	True	TP (True Positive)	FN (False Negative)
	False	FP (False Positive)	TN (True Negative)

Nilai *True Positive* diperoleh dari jumlah prediksi yang bernilai benar dan jumlah nilai asli yang benar untuk setiap *class* yang diuji. Nilai *False Negative* diperoleh dari jumlah prediksi yang bernilai salah dan jumlah nilai asli yang benar untuk setiap *class* yang diuji. Nilai *False Positive* diperoleh dari jumlah prediksi yang bernilai benar dan jumlah nilai asli yang salah untuk setiap *class* yang diuji. Nilai *True Negative* diperoleh dari jumlah prediksi yang bernilai salah dan jumlah nilai asli yang salah untuk setiap *class* yang diuji.

3. Sistem yang Dibangun

3.1 Gambaran Umum Sistem

Menghitung buah cabe berwarna hijau melalui beberapa proses seperti preprocessing citra : pemilihan ruang warna, filtering, segmentasi warna. Setelah preprocessing citra inputan dihitung luas areanya untuk memperkecil luas daerah ya akan dihitung. Alur sistem secara keseluruhan dapat dilihat pada gambar 3.1



Gambar 3.1 Gambaran Umum sistem secara keseluruhan

3.2 Akuisisi Gambar

Akuisisi gambar adalah langkah pertama dari penelitian ini dan kualitas gambar memainkan peran penting dalam langkah pemrosesan gambar yang akan dilakukan. Semua gambar harus diambil dalam kondisi pencahayaan yang terkendali. Sehingga, algoritma atau metode yang baik dibutuhkan untuk mengatasi masalah ini.

3.3 Pemilihan Ruang Warna

Ruang warna yang digunakan dalam segmentasi citra adalah ruang warna RGB. Komponen merah, hijau dan biru selanjutnya dipisahkan. Warna merah yang kurang atau tidak ada warna hijaunya dipisahkan, fungsinya adalah untuk *me-remove* warna yang cenderung merah saja. Selanjutnya setelah komponen hijau di *subtract* maka dideteksi jika obyek cenderung merah maka obyek dihapus.

Hasil proses daripada Pemilihan Ruang Warna di atas adalah seperti pada gambar 4.1. berikut.



Gambar 3.2 Gambar RGB, Gambar komponen merah dikurangi komponen hijau, Gambar hasil setelah warna yang cenderung merah dihapus

Warna biru yang dominan seperti warna langit dihapus dengan cara yang sama dengan warna kemerahan, yaitu mengambil komponen *Green* dan *Blue*, komponen *Blue* di *subtract* dengan komponen *Green*, hasilnya warna yang ada cenderung kebiruan. Warna yang cenderung biru selanjutnya dihapus. Hasil daripada proses adalah gambar 4.2.



Gambar 3.3 Proses menghilangkan warna yang cenderung biru

3.4 Filtering

Filter smoothing digunakan untuk reduksi *noise* dan efek kabur pada gambar. *Filter* pemulusan diterapkan untuk menghilangkan *detail* kecil dan menjembatani kesenjangan kecil dalam kontur. *Output* dari *filter* pemulusan adalah rata-rata piksel lingkungan yang dikelilingi oleh *masker filter*, efek pemulusan bergantung pada ukuran *mask filter*. *Filter smoothing* menggantikan nilai setiap piksel pada gambar dengan rata-rata tingkat intensitas di lingkungan yang ditentukan oleh *filter mask*. Suara acak dari gambar biasanya terdiri dari transisi tajam pada tingkat intensitas, sehingga *filter* pemulusan mampu melakukan pengurangan *noise* secara efektif. Setelah *trial* dan *error* di Matlab, ukuran *filter* Gaussian 3x3 memiliki performa terbaik dalam menghilangkan *noise* pada gambar *input* dan *edge of edge* objek yang diminati masih dapat dilihat seperti pada gambar 3.3. Dengan demikian, filter Gaussian 3x3 diterapkan untuk menghilangkan *noise* sebelum diproses ke langkah berikutnya.



Gambar 3.4 Perbandingan antara citra masukan dan gambar yang difilter Gaussian

3.4 Segmentasi Citra

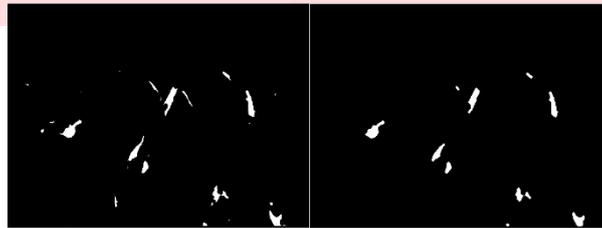
Segmentasi citra adalah proses mempartisi citra menjadi beberapa segmen untuk menemukan objek yang diminati. Ada beberapa teknik dan algoritma untuk melakukan segmentasi pada gambar digital seperti *thresholding*, *clustering*, *edge detection*, metode berbasis *histogram*, dan lain lain. Dalam proyek ini, metode *thresholding* digunakan untuk segmentasi cabe.

Citra masukan diubah menjadi ruang warna RGB dan ditunjukkan pada hijau kemerahan. Selanjutnya, gambar *grayscale* hijau kemerahan diubah menjadi gambar biner dengan *thresholding* 0,3 dengan piksel cabe berwarna putih dan piksel latar belakang berwarna hitam, seperti yang ditunjukkan pada gambar 3.5.



Gambar 3.5 Mengkonversi gambar *grayscale* menjadi gambar biner dengan *thresholding* 0,3

Wilayah tersegmentasi akan memiliki beberapa lubang atau suara di dalam wilayah citra biner pada gambar 2, fungsi "infill" digunakan untuk mengisi lubang pada citra tersegmentasi biner. Lubang dan kebisingan di dalam wilayah tersegmentasi telah terisi dengan piksel putih seperti yang ditunjukkan pada gambar 3.6.



Gambar 3.6 Gambar kiri menunjukkan lubang diisi di wilayah tersegmentasi. Gambar kanan menunjukkan erosi daerah tersegmentasi

Selanjutnya, proses erosi gambar diaplikasikan pada citra biner pada Gambar 3.6 untuk memisahkan daerah pengelompokan cabe menjadi 2 daerah yang terpisah seperti yang ditunjukkan pada gambar 3.6. Erosi gambar menggunakan *diamond mask*.

3.5 Analisis Bentuk

Analisis bentuk untuk menganalisa dan mengolah bentuk geometris di wilayah tersegmentasi yang diminati. Dalam proyek ini analisis kebulatan digunakan untuk meningkatkan akurasi deteksi objek. Luas dan keliling masing-masing daerah tersegmentasi akan diperkirakan. Koordinat batas digunakan untuk memperkirakan luas dan keliling daerah tersegmentasi yang menggunakan fungsi morfologi dengan batas yang diekstraksi. Nilai kebulatan dari daerah tersegmentasi diperoleh dengan menggunakan rumus [4].

$$R = \frac{4\pi A}{p^2} \quad (3.1)$$

dimana

R = kebulatan daerah tersegmentasi

A = luas wilayah tersegmentasi

P = perimeter wilayah tersegmentasi

3.5.1 Training LVQ

Untuk mendapatkan hasil apakah dari deteksi morfologi sesuai dengan gambar cabe atau bukan, maka sebelum proses pengenalan perlu dilakukan *training* LVQ terlebih dahulu dengan menggunakan beberapa data jenis cabe-cabe. Adapun proses yang dilakukan adalah dengan mencari bobot-bobot tersembunyi berdasarkan fitur area *boundaries*, bobot-bobot dicari dengan coba-coba sehingga menghasilkan hasil pengenalan yang mendekati sama dengan urutan file training dari fitur-fitur areanya.

Sebelum melakukan proses literasi pembobotan, terlebih dahulu parameter LVQ ditentukan. Parameter terdiri dari nilai alfa, decalfa, minalfa, makiterasi, dan iterasi. Setelah itu baru ditentukan pembobotan awal untuk melakukan iterasi selanjutnya.

Berdasarkan rumus 3.1, nilai kebulatan sama dengan 1 untuk sebuah lingkaran dan nilainya kurang dari 1 untuk bentuk lainnya. Melakukan *regionsprops* untuk menentukan area *boundingbox*, menentukan area yang lebih besar dari 1%.



Gambar 3.7 Hasil deteksi bentuk tanpa ada pembatas luas area dan Aplikasi threshold area dari hasil LVQ

Dengan memberikan batasan area yang dideteksi dari LVQ, maka area akan menyempit di sekitar area yang di-threshold oleh deteksi warna gambar saja. Hasilnya menjadi seperti gambar 4.7

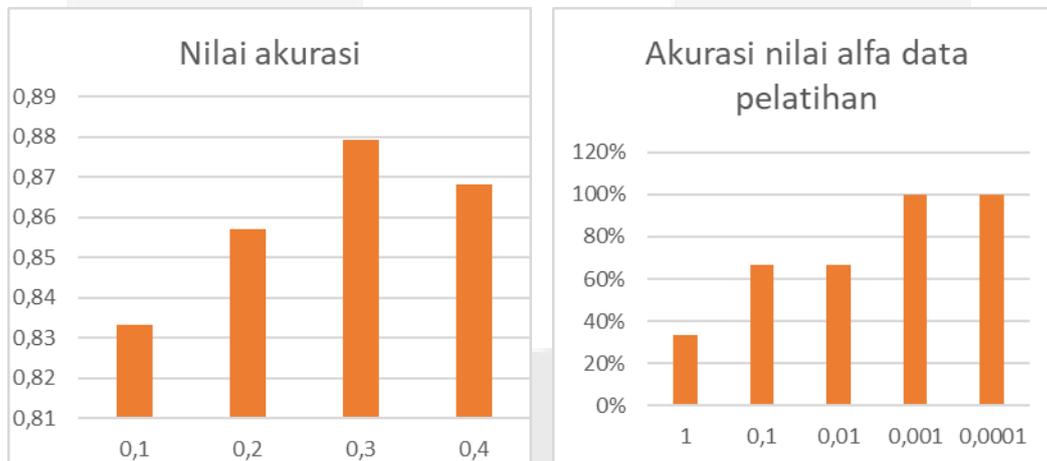
3.6 Penghitungan Cabe Otomatis

Penghitungan cabe otomatis menggunakan ambang ukuran untuk wilayah tersegmentasi. Setelah wilayah tersegmentasi lebih besar dari area ambang yang ditentukan di area total pada gambar. Setiap daerah putih yang lebih besar dari luas ambang batas akan dihitung sebagai 1 cabe yang ditemukan pada gambar. Selanjutnya, analisis bentuk akan menghitung daerah tersegmentasi dengan nilai kebulatan antara 0,45 dan 0,60 sebagai satu buah cabe lagi. Jumlah cabe total adalah jumlah jumlah buah dalam ukuran ambang batas dan analisis bentuk.

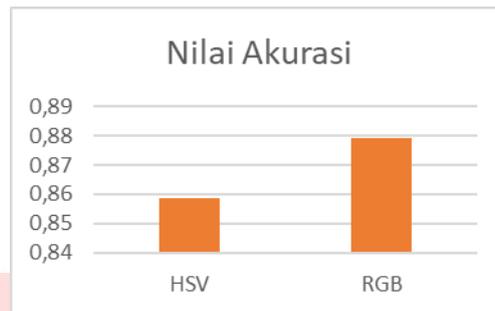
4. Evaluasi

4.1. Hasil Pengujian

Dalam proses menghitung jumlah buah cabe berwarna hijau didapat 3 hasil pengujian yaitu analisis pada nilai thresholding yang baik dengan mengukur akurasi, nilai parameter pada LVQ yaitu nilai alfa juga dilakukan pengukuran akurasi, dan analisis saat menggunakan ruang warna hsv dibandingkan hanya pemisahan warna .



Gambar 4.4.1 Gambar sebelah kiri menunjukkan nilai akurasi thresholding dan gambar sebelah kanan menunjukkan nilai akurasi parameter alfa



Gambar 4.2 Gambar perbandingan nilai akurasi pada saat citra diubah kebentuk HSV dan hanya menggunakan tranformasi ruang RGB

4.2 Analisa Hasil pengujian

Pada gambar 4.1 terlihat bahwa nilai akurasi thresholding yang tinggi pada nilai ambang batas 0,3, oleh sebab itu thresholding yang digunakan 0,3. Nilai thresholding sangat berpengaruh dalam hal ini. Terlihat juga pada pengukuran akurasi untuk akurasi nilai alfa data pelatihan. Nilai alfa merupakan parameter penting dalam training lvq atau analisis bentuk, dari grafik diatas menunjukkan bahwa semakin kecil alfa maka semakin baik dalam proses training lvq tersebut, tingkat ketelitiannya semakin baik. Analisis perbandingan citra menggunakan RGB dan HSV menunjukkan bahwa dengan menggunakan transformasi ruang warna rgb saja sudah cukup untuk mendapatkan hasil yang baik.

4.1.3 Hasil Perhitungan Cabe

Hasil perhitungan cabe dapat dilihat pada lampiran yang mengilustrasikan perbedaan antara penghitungan otomatis di ruang warna RGB dengan perhitungan secara manual. Perbedaan yang terjadi antara perhitungan manual dan perhitungan otomatis.

5. Kesimpulan

Dari pengujian yang dilakukan dapat ditarik kesimpulan bahwa dari hasil pengujian didapatkan nilai thresholding yang digunakan yaitu 0,3 dengan akurasi 87,9%. Dari hasil perhitungan akurasi parameter alfa yang baik yaitu dengan nilai 0,001 dan 0,0001 dengan akurasi 100%, jadi pada penelitian ini digunakan alfa dengan nilai 0,0001. Dengan menggabungkan nilai thresholding 0,3 dan parameter alfa 0,0001 didapatkan akurasi sebesar 87,9%.

6. Saran

Saran untuk penelitian selanjutnya mengenai perhitungan jumlah cabe berwarna hijau menggunakan transformasi ruang warna rgb yaitu melakukan penelitian dengan menggunakan data yang lebih banyak dan bervariasi. Penggunaan metode template matching mungkin dapat digunakan untuk metode selanjutnya dalam kasus ini. Menggunakan metode learning secara maksimal mungkin bisa dilakukan pada penelitian selanjutnya.

Daftar Pustaka

- [1]. Kusumanto, R., & Tompunu, A. N. (2011). Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB. *Seminar Nasional Teknologi Informasi & Komunikasi Terapan*.
- [2]. Mafrur, R. (2011). Pengenalan Huruf Jawa Menggunakan Metode Learning Vectorq Quantization. *Reaserc Gate*.
- [3]. Oktavia, A. A. (2017). Identifikasi Wajah Menggunakan Metode Local Deviralife. *Telkom University*.
- [4]. Sengupta, S., & Won, L. S. (2014). Identification and determination of the number of green citrus fruit in a canopy under different ambient light condition. *Biosystem Engineering of Elsevier*, 51-61.
- [5]. Syakri, S. A., Mulyadi, & Simbolon, Z. K. (2017). Identifikasi tingkat kebulatan buah pepaya berdasarkan luas objek dengan pengolahan citra. *Jurnal Infomedia*, 2527-9858.
- [6]. Wurtandi, M. D., & I, A. (2012). Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation dan Learning Vector Quantization. *KOMPUTA*.
- [7]. Y Song, C. G. (2014). Automatic fruit recognition and counting from multiple image. *Elsevier*, 203-215.