

---

# PENERAPAN ALGORITMA *PRINCIPLE COMPONENT ANALYSIS (PCA)* DAN FITUR RGB UNTUK PELACAKAN JENIS DAN WARNA BUAH

<sup>1</sup>Suta Wijaya, <sup>2</sup>Hendri, <sup>3</sup>Gasim

Jurusan Teknik Informatika, STMIK GI MDP, Palembang

e-mail: <sup>1</sup>aurelia\_ayah@yahoo.com, <sup>2</sup>ndriva06@gmail.com, <sup>3</sup>gasim@mdp.ac.id

## Abstrak

Saat ini banyak metode untuk menentukan jenis dan warna objek dalam suatu citra, salah satu metode untuk mengenali jenis suatu objek adalah dengan menggunakan algoritma PCA. Banyak penelitian yang menggunakan algoritma PCA hanya untuk mengenali bentuk atau jenis dari wajah seseorang, sedangkan fitur RGB adalah salah satu metode untuk mengenali warna dari sebuah objek citra. Untuk itu penelitian ini ingin membuktikan apakah algoritma PCA dan fitur RGB dapat mengenali jenis dan warna buah dalam sebuah citra. Dengan cara mengumpulkan data untuk dijadikan data training yang akan menunjang tingkat akurasi dari Algoritma PCA. Dari hasil implementasi dan pengujian Algoritma PCA dan fitur RGB, didapat kesimpulan bahwa Algoritma PCA dan fitur RGB mampu menjawab kebutuhan dalam pelacakan jenis dan warna buah dengan tingkat akurasi Algoritma PCA sebesar 86,7% sedangkan fitur RGB dapat mengenali warna dengan tingkat akurasi 100%.

**Kata kunci :** Algoritma PCA, fitur RGB

## Abstract

There are currently many methods to determine the type and color of the object in an image, one method to identify the type of an object is by using the PCA algorithm. Many studies using PCA algorithm only to recognize the shape or the type of a person's face, while the RGB feature is one method to recognize the color of an object image. For this study was to prove whether the PCA algorithm and RGB feature can identify the type and color of the fruit in an image. By collecting data to be used as training data that would support the accuracy of the PCA algorithm. From the results of the implementation and testing of the PCA algorithm and RGB features, concluded that PCA algorithm and RGB feature is able to address the needs in tracking the type and color of the fruit with an accuracy rate of 86.7% PCA algorithm while the RGB feature can recognize color with 100% accuracy rate.

**Keyword :** PCA Algorithm, RGB Feature

## 1. PENDAHULUAN

Saat ini banyak metode yang dapat digunakan untuk mengenali jenis dan warna dari sebuah objek dalam citra, salah satu metode untuk mengenali jenis sebuah objek dalam citra adalah algoritma PCA, sedangkan fitur RGB adalah salah satu metode untuk mengenali warna dari sebuah objek dalam citra. Kebanyakan penelitian yang dilakukan dengan menggunakan algoritma PCA adalah mengenali jenis dari wajah seseorang, keunggulan *Principal Component Analysis (PCA)* relatif mudah menangani sejumlah data yang cukup besar serta kemampuannya menangani data-data dimensi yang kompleks, seperti fitur-fitur pada citra wajah yang diekstrak

---

berupa dimensi *eigenvektor* dan *eigenvalues* sehingga diperoleh ruang wajah (*eigenspace*) atau *eigenfaces* [1].

Sementara kekurangannya, nilai *threshold* sangat berpengaruh dalam penentuan pengenalan pola suatu citra khususnya pada citra wajah ini, karena jika citra yang diuji mempunyai nilai fitur utama lebih besar dari nilai *threshold*, maka citra yang diuji kemungkinan bukan suatu citra yang diharapkan. Sebaliknya jika citra yang diuji mempunyai nilai fitur utama sama atau di bawah nilai *threshold*, maka citra tersebut akan teridentifikasi atau dikenali sebagai citra yang sama dengan citra yang sudah di *training* [1]. Masalah utama yang timbul adalah waktu pembangunan *principal component* yang cukup lama dan setiap kali data untuk latihan berubah maka *principal component* harus dibangun lagi [2]. Prototype pengenalan wajah melalui webcam dengan menggunakan algoritma *Principle Component Analysis* (PCA) dan *Linear Discriminant Analysis* (LDA) masih belum dapat mengenali wajah dengan benar [3]. Sistem memiliki keterbatasan dalam pengenalan dan proses *training*, sehingga untuk mengenali image yang diharapkan, pengguna harus melakukan penginputan data dari awal lagi [4].

Untuk itu penelitian ini ingin membuktikan apakah algoritma PCA juga dapat mengenali jenis dari buah dalam sebuah citra, banyak penelitian yang berkaitan dengan buah, keunggulan dari penelitian mengenai buah adalah Informasi yang dihasilkan dapat digunakan sebagai alternatif pakar dalam berkonsultasi tentang penyakit pascapanen yang meliputi nama penyakit, gejala, penyebab, probabilitas, dan cara pengendaliannya [5].

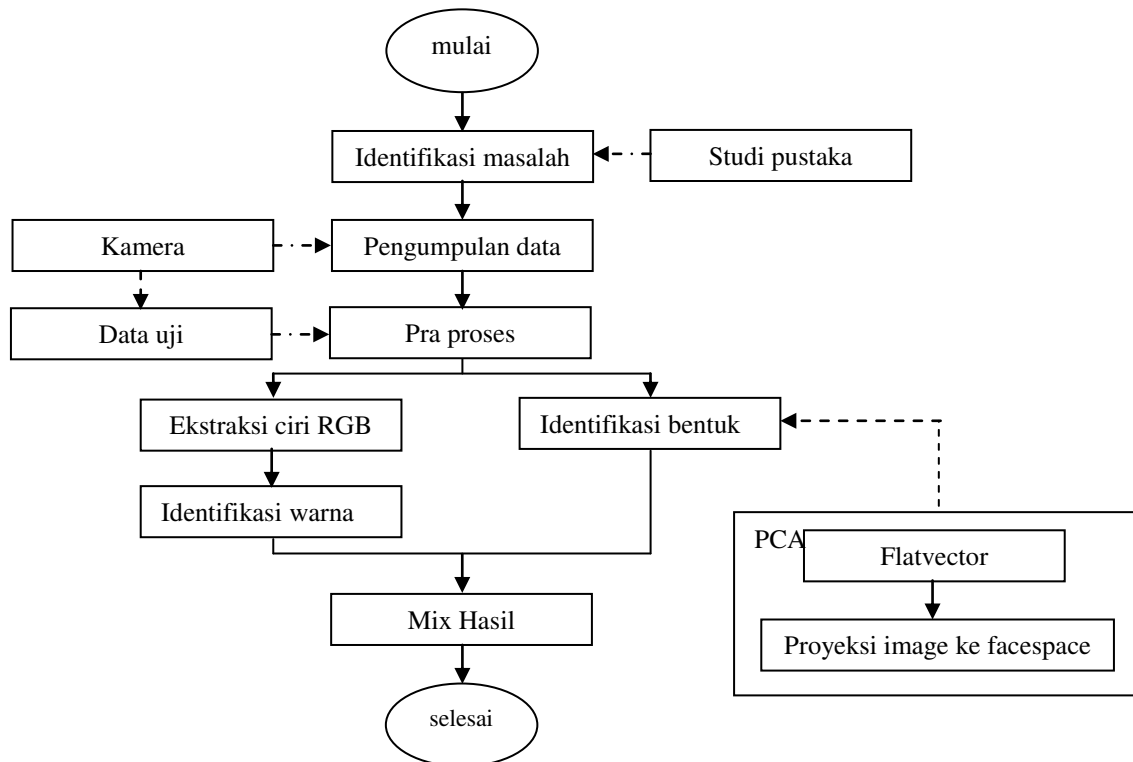
Sementara kekurangan, jarak antar buah harus minimal 25 cm agar proses dapat berjalan dengan baik [6]. Pencahayaan sangat berpengaruh [6]. Program yang telah dibuat hanya mampu membandingkan kemiripan warna dua buah citra [7].

Dari beberapa penelitian terdahulu yang terkait dengan penelitian ini, maka akan dibuktikan apakah algoritma PCA dan fitur RGB dapat mengenali jenis dan warna buah dalam suatu citra. Penelitian ini mencari dan menandai jenis dan warna buah tertentu pada sekelompok buah, citra yang didapat akan di olah menggunakan algoritma PCA untuk mengidentifikasi jenis objek pada citra yang telah di dapat, lalu citra akan di proses dengan fitur RGB untuk mengetahui warna dari objek tersebut.

---

2. METODE PENELITIAN

Tahapan-tahapan dari penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Metodologi Penelitian

2.1 Identifikasi Masalah

Pada tahap ini, proses mengumpulkan berbagai jurnal dari sumber yang berbeda dimana pada jurnal yang dikumpulkan mengacu pada orientasi Algoritma *Principle Component Analysis* yang telah dipakai pada penelitian terdahulu.

2.2 Pengumpulan Data

Pengumpulan data jenis dan warna buah dengan jenis buah apel, mangga dan jeruk. Pengumpulan data di lakukan dengan sampling buah sebanyak 75 citra (25 citra apel, 25 citra mangga, dan 25 citra jeruk) dengan menggunakan kamera dengan resolusi 7.1 megapixel.

Buah yang digunakan adalah buah yang sudah matang (sudah tua dan sudah sampai waktunya untuk dipetik) dan Objek buah yang diambil gambarnya tidak dalam keadaan sedang terbungkus plastik dan keadaannya harus utuh. Pengambilan citra dilakukan pada ruang terbuka dan pada saat siang hari, proses ini terlihat pada Gambar 2.



**Gambar 2. Proses Pengumpulan Data**

### 2.3 Pra Proses

Pada tahap ini proses melakukan pemilihan citra dengan kualitas terbaik (Gambar 2.a) dari citra yang telah dikumpulkan setelah itu dilakukan proses *cropping image* (Gambar 2.c) dan melakukan pemberian identitas dengan mengganti nama (*rename*) citra yang akan di olah sehingga mudah untuk dikenali (Gambar 2.b) pada citra yang akan di dimasukkan ke dalam *database* yang akan digunakan sebagai data training pada saat pelacakan objek. Proses ini terlihat dari Gambar 3 yang meliputi pemilihan kualitas citra terbaik, pemberian identitas citra, dan *cropping* citra.

#### a. Pemilihan Kualitas Citra

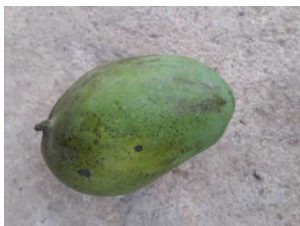


Kualitas gambar kurang Baik

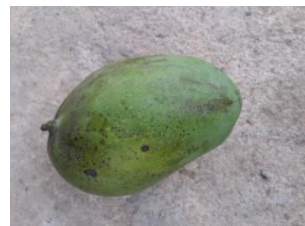


Kualitas gambar baik

#### b. Pemberian Identitas Citra



20150104\_173822.jpg



Mangga01.jpg

c. Cropping Citra



Sebelum *Cropping*  
ukuran 3096 x 4126  
pixel



Sesudah *Cropping*  
ukuran 200 x 200 pixel

**Gambar 3. Pra Proses**

**2.4 Identifikasi Bentuk**

Tahapan ini merupakan tahap proses pembentukan data training yang nantinya akan dijadikan data tes terhadap citra yang akan dikenali, tahapan-tahapan yang dilakukan adalah *flatvector* dan *facespace*.

**2.4.1 Flatvector**

Citra RGB ubah menjadi citra hitam putih dengan melakukan *treshold* sehingga nantinya akan didapat citra biner (0 dan 1) secara matematis dapat dihitung [8]:

$$T = \frac{F_{max} + F_{min}}{2} \dots\dots\dots (1)$$

Dimana :

- T = nilai Treshold
- Fmax = nilai intensitas maksimum pada citra
- Fmin = nilai intensitas minimum pada citra

Jika  $f(x,y)$  adalah nilai intensitas pixel pada posisi  $(x,y)$  maka pixel tersebut diganti putih atau hitam tergantung kondisi berikut [8] :

- $f(x,y) = 1$ , jika  $f(x,y) \geq T$
- $f(x,y) = 0$ , jika  $f(x,y) < T$

contoh :

200	230	150
240	50	170
210	100	120

Maka nilai T didapat :

$$T = \frac{240 + 50}{2} = \mathbf{145}$$

Maka di peroleh citra :

1	1	1
1	0	1
1	0	0

dengan menggunakan bahasa program matlab :

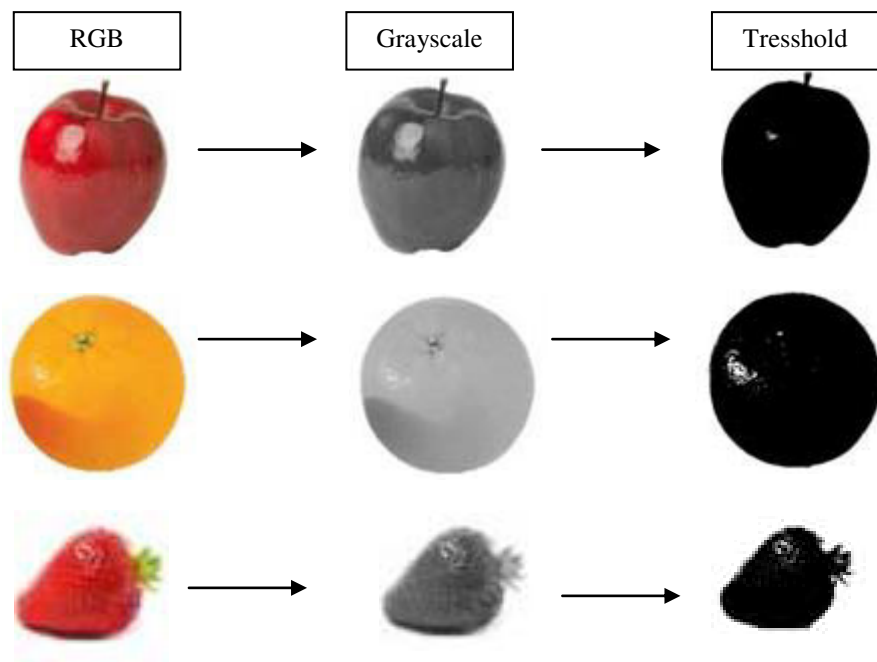
```
thresh = im2bw;
```

Contoh :

```
apel1 = imread('CA1.jpg');  
thresh = im2bw(apel1);
```

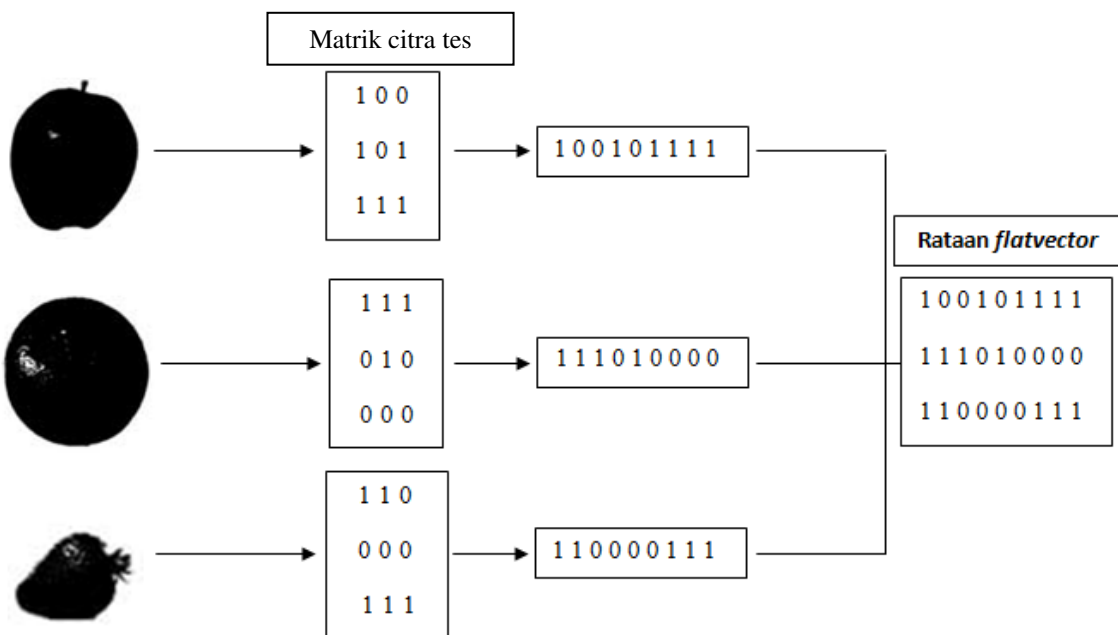
Proses ini terlihat pada Gambar 4.

---



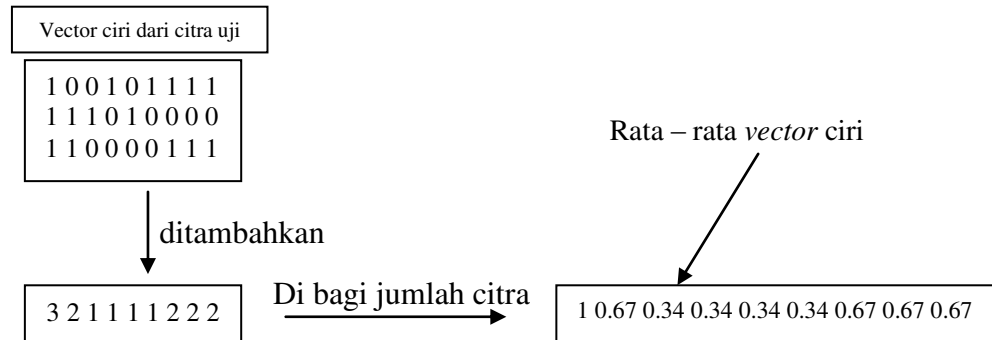
**Gambar 4. Gambar RGB di Ubah Menjadi Treshold**

Selanjutnya adalah mengubah ukuran (ordo) matrik tiap citra biner menjadi suatu matriks tunggal. Misalnya *image* yang disimpan berukuran  $H \times W$  piksel dan jumlahnya  $N$  buah, maka akan dimiliki *vector* ciri dengan dimensi  $N \times (W \times H)$ , *vector* ciri adalah suatu format gambar yang disusun berdasarkan *vector* dengan tujuan meminimalisasi penggunaan tempat penyimpanan data. Misalnya dalam data *training* terdapat 3 *image* dengan ukuran  $3 \times 3$  piksel maka kita akan mempunyai *eigenvector* ukuran  $3 \times 9$  [9]. Simulasinya dapat dilihat pada Gambar 5.



**Gambar 5. Penyusunan Flatvector**

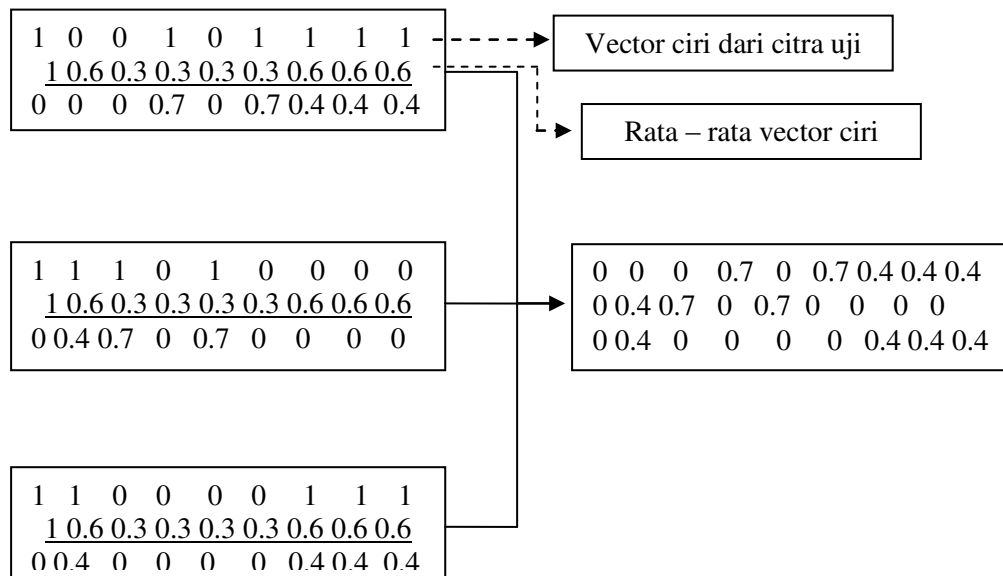
Dari *vector* ciri yang telah diperoleh, jumlahkan seluruh barisnya sehingga diperoleh matriks berukuran  $1 \times (W \times H)$ . Setelah itu bagi matriks tersebut dengan jumlah image  $N$  untuk mendapatkan nilai rata-rata *vector* ciri [9], proses ini dapat dilihat pada Gambar 6.



Gambar 6. Rataan *Flatvector*

#### 2.4.2 *Facespace*

Proyeksikan *image* ke *facespace*, *Image* diproyeksikan ke *facespace* dengan mengkalikan di basis *eigenface*. Proyeksi *vector* wajah akan dibandingkan dengan *vector* yang sesuai. Dengan menggunakan nilai rata-rata *vector* ciri, akan dihitung *eigenface* untuk matriks *vector* ciri yang telah disusun. Caranya dengan mengurangi baris-baris pada matriks *vector* ciri dengan nilai rata-rata *vector* ciri. Jika didapatkan nilai di bawah nol, ganti nilainya dengan nol [9], proses ini dapat dilihat pada Gambar 7.



Gambar 7. *Facespace*

Lakukan pengurangan baris-baris pada matriks *vector* ciri dengan nilai rata-rata *vector* ciri. Jika didapatkan nilai di bawah nol, ganti nilainya dengan nol, maka di dapatlah *facespace* untuk citra tersebut [9].



## 2.5 Identifikasi Warna

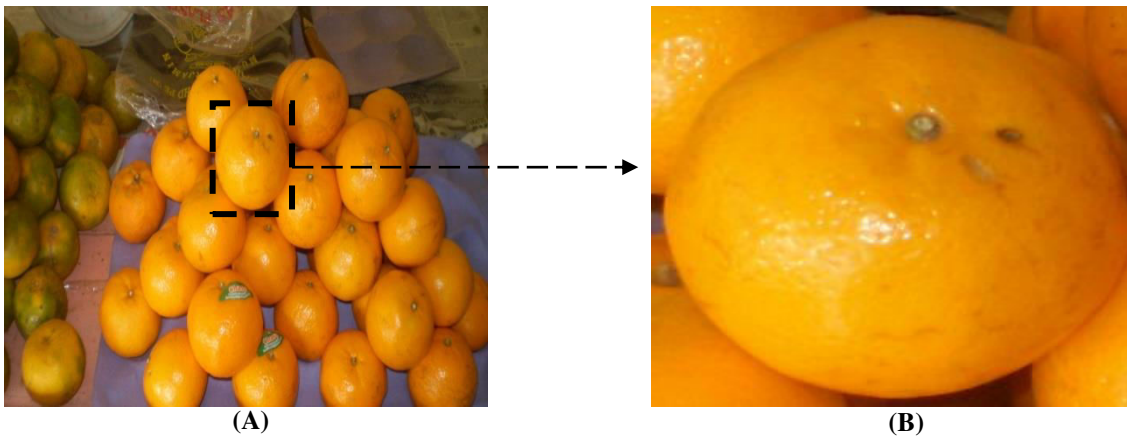
Pada tahapan ini dilakukan pengenalan warna dengan pengambilan nilai RGB, R (0-255), G (0-255), B(0-255) dimana R = 8bit, G = 8bit, dan B =8bit, menggunakan bahasa program dengan menggunakan aplikasi matlab :

```
R=B(:,:,1); lalu sum(sum(R))
```

```
G=B(:,:,2); lalu sum(sum(G))
```

```
B=B(:,:,3); lalu sum(sum(B))
```

Setelah citra didapat maka selanjutnya dilakukan proses *cropping*, setelah itu di proses dengan menggunakan bahasa program matlab untuk mendapatkan nilai RGB yang nantinya akan digunakan untuk mengetahui warna dari objek tersebut, proses cropping dapat dilihat pada Gambar 8.



**Gambar 8. Cropping Gambar**

Dengan menggunakan matlab maka didapat nilai :

R = 255

G = 246

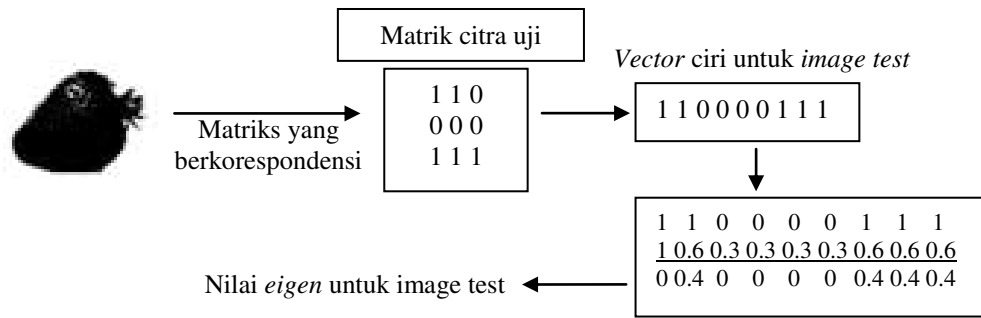
B = 178

## 3. HASIL DAN PEMBAHASAN

Pada tahapan ini didapatkan hasil yang terdiri dari PCA, dan RGB.

### 3.1 PCA

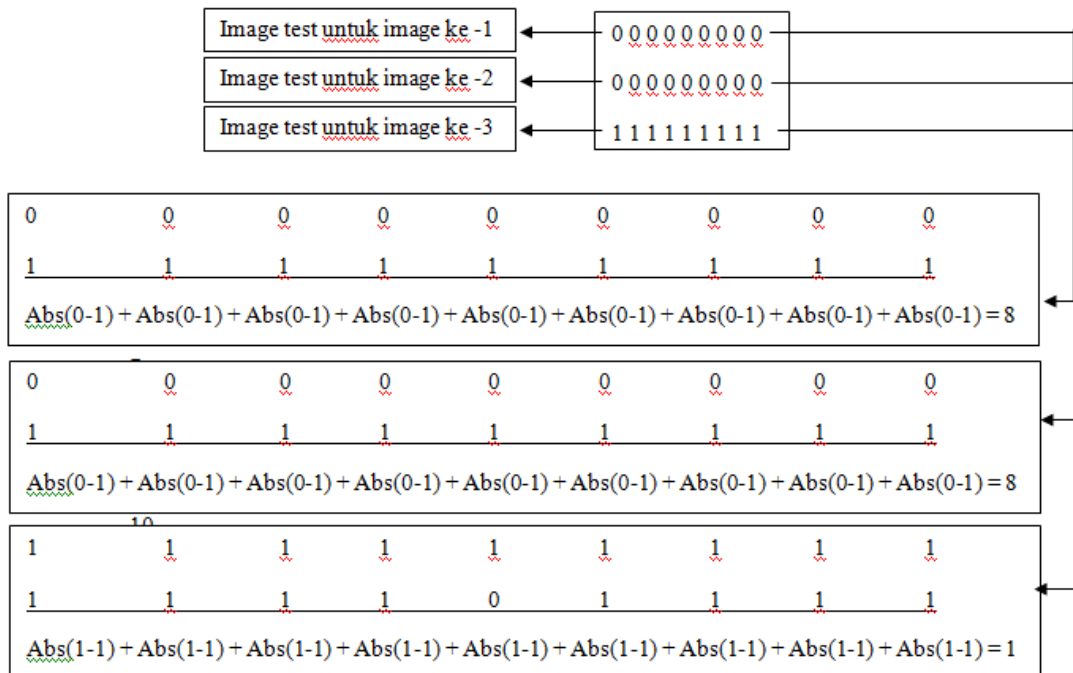
Hasil proyeksi tersebut diekstraksi dengan perhitungan PCA untuk mendapatkan feature dari image. Feature adalah komponen-komponen penting dari image-image training seperti matrik *image*. Feature inilah yang nanti akan digunakan untuk mengidentifikasi image yang akan dikenali. Kalkulasi nilai eigenface untuk matriks testface, dengan cara yang sama dengan penentuan eigenface untuk vector ciri [9], proses ini terlihat pada Gambar 9.



**Gambar 9. Matrik Image Test**

Selanjutnya ubah matrik citra uji menjadi *vector* ciri citra uji, kemudian kurangkan dengan nilai rata – rata *vector* ciri dari citra data training, apabila hasilnya di bawah nol maka gantikan nilai nya dengan nol [9].

Kemudian bandingkan nilai *euclidean distance* minimum dari *image* yang di *capture* dengan *image* yang sudah ada di *database*. Setelah nilai *eigenface* untuk *image test* diperoleh maka kita bisa melakukan identifikasi dengan menentukan jarak (distance) terpendek dengan *eigenface* dari *eigenvector training image*. Caranya dengan menentukan nilai absolute dari pengurangan baris i pada matriks *eigenface training image* dengan *eigenface* dari *testface*, kemudian jumlahkan elemen-elemen penyusun *vector* yang dihasilkan dari pengurangan tadi dan ditemukan jarak d indeks i. Lakukan untuk semua baris. Cari nilai d yang paling kecil [9], proses ini terlihat pada Gambar 10.



**Gambar 10. Hasil Perhitungan Euclidean Distance**

Karena jarak *eigenface image* ke 3 dengan *image test* yang paling kecil, maka hasil identifikasi menyimpulkan bahwa *image test* lebih mirip *image* ke 3 dari pada *image* ke 1 dan 2.

Sehingga didapat hasil dari percobaan dengan data uji sebanyak 15 (lima belas) data, terlihat pada Tabel 1.

Tabel 1. Nilai Seluruh *Euclidean Distance*

Data Training

	Apel1	Apel2	Apel3	Jeruk1	Jeruk2	Jeruk3	Mangga1	Mangga2	Mangga3	Keterangan
CA1	331.889	<del>167.778</del>	453.89	722	504.33	691.44	673	270.56	518.33	Akurat
CA2	<del>167.78</del>	337.78	466.56	678.22	483.22	625.89	647.22	289.89	536.56	Akurat
CA3	453.89	<del>466.56</del>	444.22	799	614	814.89	545.78	<del>322.22</del>	376.89	<del>Tidak Akurat</del>
CA4	647.89	<del>167.778</del>	453.89	722	504.33	691.44	673	270.56	518.33	Akurat
CA5	<del>167.78</del>	571.56	466.56	678.22	483.22	625.89	647.22	289.89	536.56	Akurat
CJ1	722	678.22	799	321.83	345.89	<del>224.11</del>	739.22	745	802	Akurat
CJ2	504.33	483.22	614	345.89	294	<del>297.11</del>	578.22	498.67	603.33	Akurat
CJ3	691.4	625.9	814.9	<del>224.1</del>	297.1	486.8	748	746.2	806	Akurat
CJ4	722	678.2	799	474.3	345.9	<del>224.1</del>	739.2	745	802.3	Akurat
CJ5	504.33	483.22	614	345.89	502.67	<del>297.11</del>	578.22	498.67	603.33	Akurat
CM1	673	647.2	545.8	739.2	578.2	748	474.8	477.1	<del>360.7</del>	Akurat
CM2	<del>270.6</del>	289.9	322.2	745	498.7	746.2	477.1	331.7	340	<del>Tidak Akurat</del>
CM3	518.3	536.6	376.9	802.3	603.3	806	360.7	<del>340</del>	785.4	Akurat
CM4	673	647.2	545.8	739.2	578.2	748	463.7	477.1	<del>360.7</del>	Akurat
CM5	518.3	536.6	376.9	802.3	603.3	806	360.7	<del>340</del>	731.7	Akurat

image test(gambar uji)

Dari Tabel 1 terlihat nilai *Euclidean distance* yang terkecil, lalu citra yang di uji dibandingkan dengan citra data training yang memiliki nilai *Euclidean distance* yang terkecil apabila sama maka citra dikenali apabila tidak sama maka citra tersebut tidak dikenali, bagian yang di beri tanda silang adalah citra data training yang memiliki nilai *Euclidean distance* terkecil setelah dilakukan perhitungan dengan citra uji (*image test*).

### 3.2 RGB

Proses selanjutnya adalah untuk mengetahui warna dari buah tersebut, proses ini dilakukan dengan cara mengambil nilai RGB dari citra uji (*image test*), citra uji terlihat pada Gambar 11.



Gambar 11. Citra Uji (*Image Test*)

Hitung nilai rata-rata dari komponen R, G, dan B, dengan menggunakan matlab, maka didapat nilai :

- R = 255
- G = 246
- B = 178

Dari hasil tersebut konversi bilangan RGB (255,246,178) menjadi bilangan hexadecimal (#FFF6B2), setelah di dapat bilangan hexadecimal maka konversi menjadi warna, warna yang di dapat dari konversi bilangan hexadecimal tersebut adalah warna *orange*.

#### 4. KESIMPULAN

Beberapa kesimpulan yang dapat diberikan adalah :

1. Algoritma PCA dan Fitur RGB dapat digunakan untuk mengenal jenis dan warna buah.
2. Berdasarkan hasil perhitungan, algoritma PCA dapat mengenali jenis dengan tingkat akurasi kemiripan jenis buah sebesar 86.7 % dari citra yang di uji.
3. Berdasarkan hasil perhitungan, terdapat 13.3% dari citra yang di uji tidak akurat dalam pengenalannya, dikarenakan nilai euclidean distance yang terpendek tidak sesuai dengan citra yang di uji.
4. Fitur RGB dapat mengenali semua warna citra yang di uji dengan persentasi 100%.

#### 5. SARAN

Dalam pengerjaan penerapan Algoritma PCA dan Fitur RGB untuk pelacakan jenis dan warna buah penulis menyadari masih banyak kekurangan pada perhitungan ini, sehingga penulis menyarankan beberapa hal, yaitu:

1. Untuk pengembangan lebih lanjut perhitungan dapat dikembangkan dengan menambahkan jumlah data training agar persentasi akurasi lebih maksimal.
2. Untuk pengembangan selanjutnya di sarankan untuk proses *tracking* bentuk dan warna dilakukan secara otomatis.

#### DAFTAR PUSTAKA

- [1] Nugraheny, Dwi 2013, *Hasil Ekstraksi Algoritma Principal Component Analysis (PCA) Untuk Pengenalan Wajah*, Teknik Informatika STTA, Yogyakarta.
  - [2] Putra, Darma 2010, *Pengolahan Citra Digital, Andi offset*, Yogyakarta.
  - [3] E.Purwanto, Jemmy, *Prototype Pengenalan Wajah Melalui Webcam dengan Menggunakan PCA dan LDA*, Jurusan Teknik Informatika Universitas Komputer Indonesia, Bandung.
  - [4] Mudrova, M, Prochazka, A, *PCA in Image Processing, Institute of Chemical Technology Prague Departement of Computing and Control Engineering, Technick'a 1905, Prague 6, Czech Republic*.
  - [5] Wijayanti Reni, Winiarti Sri 2013, *Sistem Pakar Mendiagnosa Penyakit pada Buah-Buahan Pascapanen*, Jurnal Sarjana Teknik Informatika Volume 1 Nomor 1, Yogyakarta.
  - [6] Leonardo Indrotanoto, Thiang 2008 , *Otomasi Pemisah Buah Tomat Berdasarkan Ukuran dan Warna Menggunakan Webcam Sebagai Sensor*, Seminar Nasional Ilmu Komputer dan Aplikasinya, Surabaya.
-

- [7] Iswahyudi, Catur 2010, *Prototype Aplikasi untuk Mengukur Kematangan Buah Apel Berdasarkan Kemiripan Warna*, Jurnal Teknologi Volume 3 Nomor 2, Yogyakarta.
  - [8] Sianipar R.H 2013, *Pemrograman MATLAB*, Informatika Bandung, Bandung.
  - [9] Ramdan, Deni 2011, *Face Recognition*, Unikom, Bandung.
-