

**PENGENALAN KADAR TOTAL PADAT TERLARUT PADA BUAH  
BELIMBING BERDASAR CITRA RED-GREEN-BLUE  
MENGUNAKAN PRINCIPLE COMPONENT ANALYSIS (PCA)  
SEBAGAI EKSTRAKSI CIRI DAN KLASIFIKASI K-NEAREST  
NEIGHBORHOOD (KNN)**

*Aries Maesya*

*Program Studi Ilmu Komputer Universitas Pakuan Bogor*

**ABSTRAK**

Proses pemilihan *feature* dari citra RGB (*Red-Green-Blue*) dilakukan untuk memprediksi buah belimbing yang dicirikan dengan kandungan TPT (Total Padat Terlarut). Dari *feature* terpilih, dilakukan transformasi komponen utama satu dimensi (1D-PCA) dan dua dimensi (2D-PCA) untuk mereduksi dimensi citra dengan proses pengenalan tingkat kemanisan yang dikelompokkan menjadi tiga, yaitu manis, sedang, dan asam. Dari 300 citra buah belimbing diperoleh hasil bahwa secara akurasi, teknik 1D-PCA maupun 2D-PCA memberikan hasil yang relatif sama. Namun dari segi kecepatan, 2D-PCA jauh lebih cepat dibanding 1D-PCA, khususnya pada bagian pembentukan sumbu. Model hubungan tingkat kemanisan sebagai fungsi dari nilai RGB memberikan tingkat determinasi terbesarnya 69.9%. Percobaan menunjukkan bahwa 1D-PCA maupun 2D-PCA mampu menerangkan sekitar 95% model hubungan tersebut yang dikembangkan pada ruang asal. Teknik PCA digabungkan dengan teknik klasifikasi KNN untuk pengenalan mampu mengenali buah kelompok manis dan asam dengan akurasi 100%. Sedangkan untuk kelompok sedang memiliki akurasi 83%.

*Kata Kunci : RGB, Total Padat Terlarut (TPT), PCA, KNN, Belimbing.*

**PENDAHULUAN**

Sortasi buah atau pemilihan buah berdasarkan mutu yang diinginkan memerlukan waktu relatif lama jika dilakukan secara manual atau konvensional. Salah satu cara pemilihan mutu buah adalah dengan memisahkan berdasarkan tingkat kemanisan. Buah belimbing atau disebut juga *starfruit* digolongkan pada 3 kelas mutu buah belimbing yaitu kelas super, kelas A, dan kelas B [1]. Pada kasus pemilihan mutu buah belimbing dapat dilakukan dengan cara pemeriksaan laboratorium kandungan Total Padat Terlarutnya (TPT). Analisis TPT akan menyebabkan kecacatan dan kerusakan kepada buah. Berdasarkan pada permasalahan tersebut maka diperlukan suatu metode yang mampu mengenali kelas buah belimbing sesuai dengan mutunya.

Penelitian *image processing* untuk pengklasifikasian mutu buah telah dilakukan antara lain, *probabilistic neural*

*networks* (PNN) untuk pemutuan buah belimbing manis [2]. Sortasi buah belimbing dengan Jaringan Syaraf Tiruan (JST) *multi-layer perceptron* [3]. Masing-masing menghasilkan klasifikasi sebesar 90,86% dan 90,5%. Ekstraksi ciri menggunakan *Principal Component Analysis* (PCA), pemisahan berdasarkan komponen *Red*(R), *Green*(G), dan *Blue*(B).

Teknik klasifikasi yang digunakan dalam penulisan ini menggunakan *K-Nearest Neighbor* (K-NN). KNN adalah metode klasifikasi objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Jarak yang digunakan adalah jarak *Euclidean*.

Tujuan penelitian ini yaitu melakukan Pengenalan Total Padat Terlarut (TPT) buah belimbing berdasarkan nilai RGB dengan ekstraksi ciri menggunakan PCA dan klasifikasi dengan K-NN. Ruang lingkup sistem yang dikembangkan ini adalah Data diambil dari

Pengenalan Kadar Total Padat Terlarut ..... (*Aries Maesya*)

data penelitian disertasi Departemen Keteknikan Pertanian [4], berjumlah 135 buah gambar yang terdiri dari 4 tahap pemanenan (usia 40, 50, 60, dan 70 hari), masing-masing 45 buah. Hasil keluaran sistem berupa prediksi kelas manis, sedang, dan asam.

**Principle Component Analysis (PCA)**

*Principal Component Analysis (PCA)* merupakan salah satu metode pre-processing data yang akan melakukan dekomposisi data sehingga menghasilkan koefisien-koefisien yang tidak berkorelasi. Tujuan PCA adalah menangkap variasi total pada citra latih dan merepresentasikan variasi tersebut dalam variabel-variabel yang jumlahnya lebih sedikit.

PCA dikenal juga dengan sebutan transformasi Karhunen-Loeve dan transformasi Hotteling. Berikut ini adalah algoritma PCA yang digunakan dalam penelitian ini [5].

1. Mengumpulkan data dalam sebuah matriks (*matriks ensemble*). Tiap data disimpan dalam bentuk vektor kolom. Kolom menunjukkan jumlah percobaan dan baris menunjukkan titik ciri dari tiap percobaan.
2. Membuat data normal dengan mengurangi setiap titik ciri dari data dengan *mean*. Mean tiap titik ciri dihitung dengan mencari rata-rata dari seluruh kolom data untuk setiap titik cirinya.
3. Menghitung matriks kovarian dari data normal
4. Menghitung nilai dan vektor eigen dari matriks kovarian.
5. Membuat principal component (PC). Nilai eigen disusun secara terurut menurun. Kemudian vektor eigen disusun sesuai dengan nilai *eigennya*. Vektor eigen yang tersusun itulah yang disebut sebagai PC.
6. Membentuk data baru. Data baru dihasilkan dengan mengalikan vektor

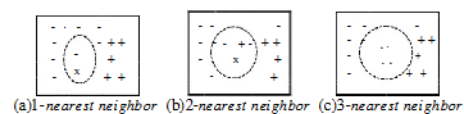
transpose dari principal component dengan data normal.

**K-Nearest Neighbour (KNN) Algorithm**

*K-Nearest Neighborhood (KNN)* adalah metode klasifikasi instance-based yang merupakan pengembangan dari 1-NN. 1-NN Dari sekumpulan objek latih, 1-NN akan memilih satu objek latih yang memiliki sifat ketetanggaan (*neighborhood*) yang paling dekat. Sifat ketetanggaan ini didapatkan dari perhitungan nilai kemiripan ataupun ketidakmiripan.

Tetangga terdekat adalah objek latih yang memiliki nilai kemiripan terbesar ataupun ketidak-miripan terkecil. Perbedaannya KNN dengan 1-NN terletak pada jumlah tetangga yang dipilih. 1-NN hanya akan memilih satu tetangga terdekat. Sedangkan KNN akan memilih K-tetangga terdekat. Dari K-tetangga itu, KNN akan menentukan hasil klasifikasi dengan melihat jumlah kemunculan dari kelas dalam K-tetangga yang terpilih. Kelas yang paling banyak muncullah yang akan menjadi kelas hasil klasifikasi. Berikut ini adalah algoritma KNN [6].

1. Menghitung nilai kemiripan atau ketidakmiripan (*similarity/dissimilarity distance*). Metode perhitungan yang digunakan ada tiga, yaitu *Euclidean distance*, *Square euclidean distance* dan *Manhattan distance*
2. Mengurutkan hasil dari perhitungan nilai kemiripan/ketidakmiripan secara terurut menurun.
3. Menentukan nilai K dan mengambil K-jumlah tetangga dari hasil langkah 2.
4. Menentukan kelas dari data uji berdasarkan kelas yang paling banyak muncul dari hasil langkah 3.



**Gambar 1:** Ilustrasi *nearest neighbour* terhadap data baru (x).

Untuk mendefinisikan jarak antara dua titik yaitu titik pada data *training* (x) dan titik pada data *testing* (y) maka digunakan rumus *Euclidean*.

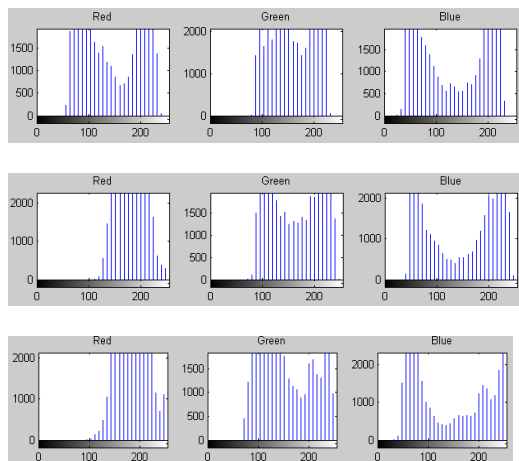
$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Dengan *d* adalah jarak antara titik pada data training x dan titik data *testing* y yang akan diklasifikasi, dimana  $x = x_1, x_2, \dots, x_i$  dan  $y = y_1, y_2, \dots, y_i$  dan *I* merepresentasikan nilai atribut serta merupakan dimensi atribut

**HASIL DAN PEMBAHASAN**

**Ekstraksi ciri**

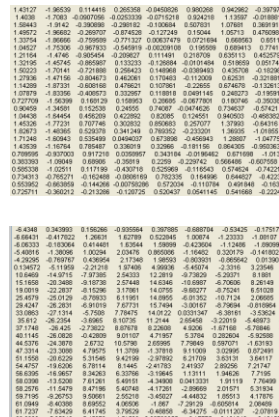
Proses awal ekstraksi ciri adalah dengan pemisahan warna *red* (R), *green* (G), dan *blue* (B), proses ini untuk mengetahui unsur warna yang lebih dominan untuk masing-masing kelas belimbing. Gambar 2 adalah contoh data testing belimbing pada kelas asam, sedang, dan manis.



**Gambar 2.** contoh RGB data testing kelas asam

Hasil pemisahan warna RGB menunjukkan bahwa unsur warna merah (*red*) memiliki nilai dominan pada kelas sedang dan manis. Proses selanjutnya adalah PCA 1 dimensi dan PCA 2 dimensi. Gambar 3 adalah contoh hasil PCA 1 dimensi dan PCA 2 dimensi untuk kelas asam.

Pengenalan Kadar Total Padat Terlarut ..... (Aries Maesya)



**Gambar 3.** Hasil PCA 1 dan 2 dimensi

**Penentuan Data Training dan Data Testing**

Seluruh data hasil praproses di bagi menjadi data *training* dan data *testing*. Data *training* digunakan sebagai masukan pelatihan menggunakan K-NN sedangkan data *testing* digunakan untuk menguji model hasil pelatihan menggunakan KNN. Jumlah data *training* sebanyak 99 gambar belimbing, masing-masing kelas yaitu asam, manis, dan sedang sebanyak 33 gambar. Data testing berjumlah 36 gambar belimbing, masing-masing kelas yaitu asam, manis, dan sedang sebanyak 12 gambar.

**Klasifikasi dengan KNN**

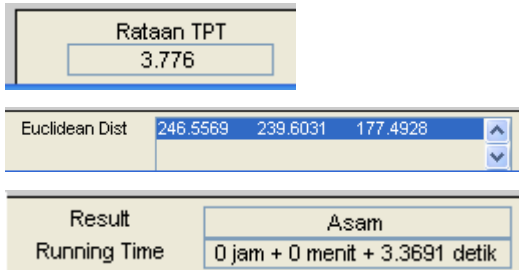
Setelah data di bagi menjadi data *training* dan data *testing*, kemudian data testing diklasifikasi menggunakan KNN.

Algoritma KNN diuraikan sebagai berikut :

1. Tentukan parameter K = jumlah tetangga terdekat.
2. Hitung jarak antara data baru dengan semua data *training*.
3. Urutkan jarak tersebut dan tetapkan tetangga terdekat berdasarkan jarak minimum ke-K.
4. Periksa kelas dari tetangga terdekat.
5. Gunakan mayoritas sederhana dari kelas tetangga terdekat sebagai nilai prediksi data baru.

Klasifikasi kelas belimbing menggunakan KNN dengan jarak euclidean. Gambar 4

adalah hasil TPT, jarak euclidean, dan hasil kelas belimbing asam dengan nilai  $K = 2$ .



**Gambar 4.** Hasil klasifikasi untuk kelas asam

**Percobaan dengan KNN**

Hasil pengujian pada 36 data testing yaitu 12 data asam, 12 data manis, dan 12 data sedang ditunjukkan pada tabel 1 berikut :

**Tabel 1.** Hasil pengujian

Data ke	Kelas Real	Prediksi Kelas
1	asam	Asam
2	asam	Asam
3	asam	Asam
4	asam	Asam
5	asam	Asam
6	asam	Asam
7	asam	Asam
8	asam	Asam
9	asam	Asam
10	asam	Asam
11	asam	Asam
12	asam	Asam
13	manis	Manis
14	manis	Manis
15	manis	Manis
16	manis	Manis
17	manis	Manis
18	manis	Manis
19	manis	Manis
20	manis	Manis
21	manis	Manis
22	manis	Manis
23	manis	Manis
24	manis	Manis
25	sedang	sedang
26	sedang	sedang
27	sedang	sedang
28	sedang	sedang
29	sedang	sedang
30	sedang	sedang
31	sedang	sedang
32	sedang	sedang
33	sedang	Sedang

34	sedang	Manis
35	sedang	Sedang
36	sedang	Manis

Hasil prediksi untuk kelas asam dan manis adalah 100% sedangkan untuk kelas sedang adalah 83%. Hasil pengujian validitas sistem untuk kelas belimbing direpresentasikan menggunakan *single decision threshold (one feature)*, fiturnya berupa kelas belimbing (asam, manis, dan sedang), dijelaskan sebagai berikut :

Prediksi

		+	-
Real	+	TP	FN
	-	FP	TN

Penjelasan TP, TN, FP, dan FN adalah :

- TP (*True Positive*), apabila nilai real dan prediksi menghasilkan nilai positif, yaitu jika :
  - Real “asam”, maka prediksi memutuskan “asam”.
  - Real “manis”, maka prediksi memutuskan “manis”.
  - Real “sedang”, maka prediksi memutuskan “sedang”.
- TN (*True Negative*), apabila nilai real dan prediksi menghasilkan nilai negatif, yaitu jika :
  - Real “tidak asam”, maka prediksi memutuskan “tidak asam”.
  - Real “tidak manis”, maka prediksi memutuskan “tidak manis”.
  - Real “tidak sedang”, maka prediksi memutuskan “tidak sedang”.
- FP (*False Positive*), apabila nilai real negatif tetapi prediksi menghasilkan nilai positif, yaitu jika :
  - Real “tidak asam”, maka prediksi memutuskan “asam”.
  - Real “tidak manis”, maka prediksi memutuskan “manis”.
  - Real “tidak sedang”, maka prediksi memutuskan “sedang”.
- FN (*False Negative*), apabila nilai real positif tetapi prediksi menghasilkan nilai negatif, yaitu jika :
  - Real “asam”, maka prediksi memutuskan “tidak asam”.

- Real “manis”, maka prediksi memutuskan “tidak manis”.
- Real “sedang”, maka prediksi memutuskan “tidak sedang”.

Perbandingan jumlah kelas asam, manis, dan sedang antara real dan prediksi disajikan pada tabel 2.

**Tabel 2.** Perbandingan jumlah kelas antara real dan prediksi

		Prediksi		
		Asam	Manis	Sedang
Real	Kelas	Asam	Manis	Sedang
	Asam	12	0	0
	Manis	0	12	0
Sedang	0	2	10	

Validitas sistem dinilai dengan cara menghitung nilai TP, TN, FP, dan FN dari tabel 2.

$$TP = 12+12+10 = 34$$

$$TN = (12+10)+(12+10)+(12+12) = 68$$

$$FP = (0+0)+(0+2)+(0+0) = 2$$

$$FN = (0+0)+(0+0)+(0+2) = 2$$

$$\begin{aligned} \text{Kinerja sistem} &= \frac{TP + TN}{TP+TN+FP+FN} \times 100\% \\ &= \frac{34+68}{34+68+2+2} \times 100\% \\ &= 96,23 \% \end{aligned}$$

## KESIMPULAN DAN SARAN

Dari hasil penelitian, dapat disimpulkan bahwa :

1. Metode ekstraksi ciri dengan PCA dengan pemisahan warna *red*, *green*, dan *blue* (RGB) dan klasifikasi dengan *K-Nearest Neighbor* (K-NN) dapat digunakan untuk mengklasifikasi kelas belimbing asam dan manis dengan nilai akurasi 100% dan kelas sedang 83%
2. Hasil keseluruhan kinerja sistem untuk prediksi kelas belimbing mencapai 96,23%

Untuk penelitian selanjutnya diharapkan kinerja sistem dapat ditingkatkan dengan menggunakan metode klasifikasi yang lain.

## DAFTAR PUSTAKA

- Departemen Pertanian, *Rancangan Standar Nasional Indonesia (RSNI) Buah Belimbing Manis* (*Averrhoa Carambola L.*), **Error! Hyperlink reference not valid.** Belimbing-1.htm [diakses pada Februari 2008], 2007.
- Zaki F, ”*Pengembangan probabilistic neural networks untuk penentuan kematangan belimbing manis*”, Skripsi Jurusan Ilmu Komputer, tidak dipublikasikan, 2009.
- Abdullah M.Z., M. Saleh J., F. Syahir, dan M. Azemi, ”*Discrimination and classification of fresh-cut starfruits (*Averrhoa carambola L*) using automated machine vision system*”, *Journal of Food Engineering*, 2005.
- Irmansyah, ”*Evaluasi mutu belimbing dengan pengolahan citra dan logika fuzzy*”, [Disertasi]. Departemen Keteknikan Pertanian, Fakultas Teknologi Pertanian IPB, Bogor, 2008.
- Jonathon Shlens, 2005, *A Tutorial on Principal Component Analysis*, California, Univeristy of California.
- Parades, Roberto dan Enrique Vidal, 2005, *Learning Weighted Metrics to Minimize Nearest - Neighbor classification Error*, Spanyol, Institut Teknologi Informatika Spanyol.