

ARTIKEL

**APLIKASI KLASIFIKASI JENIS JENIS BUAH JERUK
MENGUNAKAN METODE *K-NEAREST NEIGHBOR***



Oleh:

RIDHO ARY SUMARNO

13.1.03.02.0092

Dibimbing oleh :

- 1. Hermin Istiasih M.T., M.M.**
- 2. Mochammad Bilal S.kom., M.Cs.**

PROGAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS NUSANTARA PGRI KEDIRI

TAHUN

2017



**SURAT PERNYATAAN
ARTIKEL SKRIPSI TAHUN 2017**

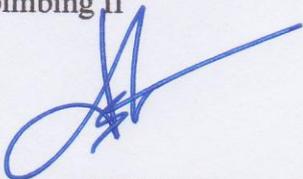
Yang bertanda tangan di bawah ini:

Nama Lengkap : Ridho Ary Sumarno
NPM : 13.1.03.02.0092
Telepon/HP : 085790475068
Alamat Surel (Email) : ridhoari74@gmail.com
Judul Artikel : Aplikasi Klasifikasi Jenis-Jenis Buah Jeruk
Menggunakan Metode K-Nearest Neighbor
Fakultas – Program Studi : Teknik / Teknik Informatika
Nama Perguruan Tinggi : Universitas Nusantara PGRI Kediri
Alamat Perguruan Tinggi : Jalan. KH. Ahmad Dahlan No.76, Mojoroto, Kota
Kediri, Jawa Timur 64112

Dengan ini menyatakan bahwa :

- artikel yang saya tulis merupakan karya saya pribadi (bersama tim penulis) dan bebas plagiarisme;
- artikel telah diteliti dan disetujui untuk diterbitkan oleh Dosen Pembimbing I dan II.

Demikian surat pernyataan ini saya buat dengan sesungguhnya. Apabila di kemudian hari ditemukan ketidaksesuaian data dengan pernyataan ini dan atau ada tuntutan dari pihak lain, saya bersedia bertanggungjawab dan diproses sesuai dengan ketentuan yang berlaku.

Mengetahui		Kediri, 10 Agustus 2017
Pembimbing I  <u>Hermin Istiasih M.T., M.M.</u> NIDN. 0014057501	Pembimbing II  <u>Mochammad Bilal S.kom., M.Cs.</u> NIDN. 0729108102	Penulis,  <u>Ridho Ary Sumarno</u> 13.1.03.02.0092

APLIKASI KLASIFIKASI JENIS – JENIS BUAH JERUK MENGUNAKAN METODE K-NEAREST NEIGHBOR

Ridho Ary Sumarno

13.1.03.02.0092

Teknik – Teknik Informatika

Email : ridhoari74@gmail.com

Hermin Istiasih M.T.,M.M¹. dan Mochamad Bilal S.kom., M.Cs².

UNIVERSITAS NUSANTARA PGRI KEDIRI

ABSTRAK

Penelitian ini dilatar belakangi bahwa tanaman Jeruk paling cocok ditanam di daerah subtropis yang memiliki suhu rata-rata 20-25°C. Jeruk merupakan buah *non-klimaerik* yaitu buah hanya dapat diperoleh di pohon atau tidak dapat diperam setelah dipanen. Identifikasi buah jeruk berdasarkan citra buah jeruk menggunakan berdasarkan kulit jeruk.

Permasalahan dari sistem klasifikasi jenis-jenis buah jeruk ini adalah bagaimana cara merancang sebuah aplikasi untuk klasifikasi buah jeruk menggunakan metode *K-Nearest Neighbor* dalam pengenalan citra buah jeruk.

Algoritma *K-Nearest Neighbor* (kadang disebut juga *K-Nearest Neighbor / K-NN*) merupakan algoritma yang melakukan klasifikasi berdasarkan kedekatan lokasi (jarak) suatu data dengan data yang lain.

Simpulan pada aplikasi klasifikasi jenis buah jeruk ini dapat menganalisis permasalahan yang ada pada sistem dapat mempercepat dan mempermudah kita untuk mengetahui perbedaan jenis - jenis jeruk dan membantu dalam melakukan klasifikasi secara integritas dan seefisien mungkin.

Saran pengembangan aplikasi klasifikasi jenis – jenis buah jeruk menjadi lebih baik yaitu menggunakan metode *K-Nearest neighbor* untuk mendapatkan nilai terdekat dari citra, selain itu, perlu adanya penggunaan metode-metode algoritma lainnya yang dapat dipakai untuk mengetahui klasifikasi buah jeruk secara lebih akurat.

Kata Kunci : klasifikasi, klasifikasi jenis-jenis buah jeruk, menggunakan metode *k-nearest neighbor*.

I. Latar Belakang

Tanaman Jeruk (*Citrus aurantium L.*) paling cocok ditanam di daerah subtropis yang memiliki suhu rata-rata 20-25°C. Jeruk merupakan buah *non-klimatik* yaitu buah hanya dapat diperoleh di pohon atau tidak dapat diperam setelah dipanen. Jadi penting halnya untuk memetik ataupun memanen buah jeruk.

Warna adalah salah satu atribut yang berperan dalam mengidentifikasi objek tertentu, pemrosesan warna termasuk didalamnya adalah ekstraksi informasi tentang *spectral properties* dari permukaan objek dan mencari kesamaan terbaik dari sekumpulan deskripsi yang telah diketahui untuk melakukan pengenalan.

Pengenalan objek citra akan diklasifikasi dengan metode pembelajaran menggunakan metode *K-NN*. Warna merupakan *respon psychophysiological dan intensitas* yang berbeda. Persepsi warna dalam pengolahan citra tergantung pada tiga faktor, yaitu : *spectral reflectance* (menentukan bagaimana suatu permukaan memantulkan warna), *spectral content*

(kandungan warna dari cahaya yang menyinari permukaan) dan *spectral response* (kemampuan merespon warna dari sensor dalam imaging system).

II. METODE

A. Grayscale

Citra skala keabuan memberi kemungkinan warna yang lebih banyak daripada citra *biner*, karna ada nilai-nilai lain diantara nilai minimum (biasanya = 0) dan nilai maksimumnya. banyaknya kemungkinan nilai dan nilai maksimumnya tergantung pada jumlah *bit* yang digunakan. Contohnya untuk skala keabuan 4 bit, maka jumlah kemungkinan nilainya adalah $2^4 = 16$, dan nilai maksimumnya adalah $2^4 - 1 = 15$ sedangkan untuk skala keabuan 8 bit, maka jumlah kemungkinan nilainya adalah $2^8 = 256$. format citra ini disebut skala keabuan karena pada umumnya warna yang dipakai adalah antara warna hitam sebagai warna minimal dan warna putih sebagai warna maksimal, sehingga warna antaranya adalah warna abu-abu. Sebagai contoh dipilih warna minimalnya adalah putih dan warna maksimalnya

adalah merah, maka semakin besar nilainya semakin besar pula.

A. Operator Sobel

Operator Sobel adalah salah satu operator yang menghindari adanya perhitungan gradient di titik interpolasi. Operator ini menggunakan kernel ukuran 3x3 pixel untuk perhitungan *gradient* sehingga perkiraan gradient berada tepat di tengah jendela. Susunan pixel-pixel disekitar pixel (x,y) dalam operator *sobel* adalah sebagai berikut

$$\begin{matrix} a_0 & a_1 & a_2 \\ a_7 & (x,y) & a_3 \dots\dots(2.3) \\ a_6 & a_5 & a_4 \end{matrix}$$

Berdasarkan susunan pixel tetangga tersebut, besaran gradient yang dihitung menggunakan operator sobel adalah sebagai berikut

$$S_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

$$S_y = (a_2 + ca_3 + a_4) \dots\dots(2.4)$$

Di mana c adalah konstanta yang bernilai 2. Sx dan Sy diimplementasikan menjadi kernel berikut :

$$-1 \ 0 \ 1$$

$$S_x = \begin{matrix} -2 & 0 & 2 \\ -1 & 0 & 1 \dots\dots(2.5) \end{matrix}$$

$$S_y = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

Arah tepi dihitung menggunakan persamaan berikut :

$$a(x,y) = \tan^{-1} \left(\frac{S_y}{S_x} \right) \dots\dots(2.6)$$

Tampak bahwa operator *Sobel* menggunakan pembobotan pada *pixel-pixel* yang lebih dekat dengan titik pusat *kernel*. Oleh karena itu, pengaruh *pixel-pixel* tetangga akan berbeda sesuai dengan letaknya terhadap titik dimana *gradient* dihitung. dalam melakukan perhitungan *gradient*, operator ini merupakan gabungan dari posisi mendatar dan posisi vertikal.

Cara kerja operator sobel adalah :

- 1) Menentukan nilai x dan y.
- 2) Menghitung nilai kernel x dan y.
- 3) Menghitung nilai Gradient.
- 4) Susun posisi x dan y yang baru.

B. K-Nearest Neighbor

Algoritma *K-Nearest*

Neighbor (kadang disebut juga *K-Nearest Neighbor / K-NN*) merupakan algoritma yang melakukan klasifikasi berdasarkan kedekatan lokasi (jarak) suatu data dengan data yang lain. Pada algoritma *K-NN*, data berdimensi N , dapat dihitung jarak dari data tersebut ke data yang lain, nilai jarak ini yang digunakan sebagai nilai kedekatan/ketidak miripan antara data uji dengan data latih. Nilai K pada *K-NN* berarti K -data terdekat dari data uji. Pada algoritma *K-NN*, sebuah data uji $z = (x', y')$, dimana x' ada vektor/atribut data uji, sedangkan y' adalah label kelas data uji yang belum diketahui, kemudian menghitung jarak (atau kemiripan) data uji ke setiap data latih $d(x', x)$, kemudian mengambil K tetangga terdekat pertama dalam Dz . Setelah itu dihitung jumlah data yang mengikuti kelas yang ada dari K tetangga tersebut. Kelas dengan data terbanyak yang mengikutinya menjadi kelas pemenang yang diberikan sebagai label kelas pada data uji y' .

C. Simulasi Algoritma

1. Deteksi tepi sobel

Misalkan Diketahui sebuah citra dengan ukuran 3×3 px, dengan citra 3 bit.

5	3	4
6	4	2
2	1	3

SX

-1	0	1
-2	0	2
-1	0	1

SY

-1	-2	-1
0	0	0
1	2	1

2. Algoritma K-NN

Mencari nilai terdekat dengan menghitung hasil dari nilai sobel dengan menggunakan algoritma *K-NN*. Data dari nilai sobel yang dihitung yaitu menggunakan nilai sobel SX, sebagai berikut :

Diketahui Data testing SX :

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Diketahui Piksel :

$$\begin{bmatrix} 5 & 3 & 4 \\ 6 & 4 & 2 \\ 2 & 1 & 3 \end{bmatrix}$$

III. HASIL DAN KESIMPULAN

A. Tampilan Program

Berikut merupakan form pada aplikasi klasifikasi buah jeruk:

1. Tampilan Awal



Gambar 5.9 Form menu
Keterangan dari gambar 5.9 adalah sebagai berikut :

1. Data *Training* : didalamnya terdapat pilihan proses dari data *training*.
2. Data *Testing* : didalamnya terdapat pilihan proses dari data *testing*.
3. Informasi : untuk mengetahui informasi aplikasi.
4. Keluar : untuk keluar dari aplikasi.

2. Form Training



Gambar 5.10 Form Data Training

Keterangan dari gambar 5.10 adalah sebagai berikut :

1. Combo Box : untuk memilih jenis jeruk yang di inginkan.
2. Input Citra: untuk mengambil foto didalam folder .
3. *Grayscale* : untuk pengkonversian citra warna RGB menjadi citra berwarna abu-abu.
4. Deteksi Tepi : untuk mengidentifikasi garis batas dari suatu objek yang terdapat pada citra.
5. Informasi Citra Masukan : untuk mrngetahui ukuran citra foto 150 x 150.
6. Reset : menghapus semua masukan yang ada didalam aplikasi.
7. Memo : mengetahui piksel nilai citra dari deteksi tepi.

8. Data *Testing* : menuju proses data *testing*.

3. Form Data Testing



Gambar 5.11 Form Data Testing

Keterangan dari gambar 5.11 adalah sebagai berikut :

1. input Citra: untuk mengambil foto didalam folder .
2. *Grayscale* : untuk pengkonversian citra warna RGB menjadi citra berwarna abu-abu.
3. Deteksi Tepi : untuk mengidentifikasi garis batas dari suatu objek yang terdapat pada citra.
4. Nilai *Eucludient* : Mengetahui nilai jarak citra yang terdekat.
5. Identifikasi : untuk mengetahui hasil citra yang telah di proses.
6. Reset : menghapus semua masukan yang ada didalam aplikasi.

7. Memo : mengetahui nilai piksel citra dari deteksi tepi.

8. Home : menuju tampilan halaman utama.

B. Hasil Uji Coba

Dari hasil pengujian menggunakan 6 buah sampel buah jeruk meliputi nipis, keprok, purut, lemon, satsuma, siam yang masing-masing berukuran 150x150 piksel.

Tabel 5.3 Hasil skenario uji coba

Skenario uji coba	Hasil		Akurasi
	Benar	Salah	
1.	28	16	63%
2.	30	24	55%
3.	25	21	54%
4.	35	25	58%

Keterangan dari tabel 5.3 hasil skenario uji coba dapat dijelaskan bahwa hasil dari proses *training* dan proses *testing* dari masing-masing skenario adalah sebagai berikut :

- a. **Skenario 1** : menghasilkan 28 bernilai benar dan 16 bernilai salah akurasi sebanyak 63%

- b. **Skenario 2** : menghasilkan 30 bernilai benar dan 24 bernilai salah akurasi sebanyak 55%
- c. **Skenario 3** : menghasilkan 25 bernilai benar dan 21 bernilai salah akurasi sebanyak 54%
- d. **Skenario 4** : menghasilkan 35 bernilai benar dan 25 bernilai salah akurasi sebanyak 58%

Lukman Adlin Harahap. 1994. *Budidaya Tanaman Jeruk*. Yogyakarta: Penerbit kansius.

Tan, P.DKK. 2006. *Introduction to Data Mining*. Boston San Fransisco New York: Pearson Education.

IV. DAFTAR PUSTAKA

Abdillah, leon, A. 2005. Universitas Bina Darma, Palembang. *Pemrograman III (Delphi Database)*.

Ahmad, Balza. 2005. Dalam T. P. DELPHI. Yogyakarta: Andi publishing.

Arham Z, U. 2004. Pascasarjana IPB. *Evaluasi mutu jeruk dengan pengolahan citra digital dan jaringan saraf digital*, 1.

Charu. G, Sunanda. G. 2013. International Journal of Science and Research. *Edge Detection of an Image based on Ant Colony Optimization Technique*, Vol.2(Issue 6, hal 114-120).

Ian H Witten, f., & E. 2011. *Data Mining: Practical Machine Learning Tools and Techniques (3 ed.)*. (A. S. Burlington, Ed.). United States of America: Morgan Kaufmann.