

DETEKSI KEBUTUHAN NITROGEN UNTUK PENENTUAN JUMLAH PUPUK PADA TANAMAN PADI BERDASARKAN WARNA DAUN MENGGUNAKAN *SUPPORT VECTOR MACHINE*

Gumilar Fajar Darajat¹, Irfan Maliki²

^{1,2} Teknik Informatika-Universitas Komputer Indonesia

Jln. Dipatiukur No. 112-116, Lebakgede, Coblong, Kota Bandung, Jawa Barat 40132

E-mail : gumilarfajardarajat@gmail.com¹, irfan.maliki@email.unikom.ac.id

ABSTRAK

Nitrogen merupakan salah satu unsur hara yang berperan penting dalam proses pertumbuhan padi. Untuk mengetahui kebutuhan nitrogen pada padi bisa dilihat dari warna daunnya. Semakin hijau warna daun pada padi maka kandungan nitrogennya semakin tinggi, sebaliknya jika warna daun padi semakin menguning maka kandungan nitrogennya semakin rendah. Sistem deteksi kebutuhan nitrogen padi berdasarkan warna daun dengan menggunakan K-Nearest Neighbor (KNN) dan *preprocessing* yang dilakukan dengan *menzoom* bagian daun padi tertentu menghasilkan nilai akurasi yang kurang baik yaitu sebesar 66,7%. Oleh karena itu penelitian ini menggunakan *machine learning Support Vector Machine* (SVM) sebagai klasifikasi sistem deteksi kebutuhan nitrogen padi berdasarkan warna daun. Tujuan dari penelitian ini adalah untuk mengimplementasikan metode *Support Vector Machine* pada sistem deteksi kebutuhan nitrogen padi berdasarkan warna daun. Dari hasil pengujian akurasi yang dilakukan, tingkat akurasi sistem dalam menganalisis dan merekomendasikan kebutuhan nitrogen secara rata-rata adalah 79% dan hasil akurasi berdasarkan waktu yang paling baik adalah pada pagi hari dengan nilai akurasi sebesar 88%. Berdasarkan hasil tersebut, disimpulkan bahwa algoritma SVM dapat diterapkan dalam sistem deteksi kebutuhan nitrogen padi berdasarkan warna daun.

Kata kunci : Rice, Nitrogen, Leaf Color Chart, SVM, Histogram.

1. PENDAHULUAN

Nitrogen adalah salah satu unsur hara makro utama yang terdapat pada padi. Nitrogen diperlukan oleh tanaman untuk proses fotosintesis sebagai pembuat zat hijau daun atau klorofil. Pemberian pupuk nitrogen yang sesuai dengan kebutuhan tanaman merupakan kunci dalam memperbaiki pertumbuhan tanaman untuk mendapatkan hasil yang optimal [1]. Pemberian pupuk nitrogen yang berlebihan bisa menyebabkan warna daun pada tanaman menjadi hijau gelap [2] sehingga tanaman tersebut bisa rentan terhadap serangan hama dan

penyakit. Pemberian pupuk nitrogen yang berlebihan juga dapat memperlambat pematangan gabah, jerami menjadi lunak sehingga tanaman mudah rebah dan kualitas gabah menjadi turun [1]. Sementara kekurangan Nitrogen menyebabkan pertumbuhan tanaman menjadi kerdil, akar terbatas, daun menjadi kuning karena kekurangan klorofil, produksi biomassa menjadi berkurang sehingga hasilnya menurun dan gabah cenderung mudah rontok [2]. Bagan Warna Daun (BWD) adalah sebuah alat yang berguna untuk mengetahui kadar hara N tanaman padi. Pada alat ini terdapat empat kotak skala warna daun, mulai dari hijau muda hingga hijau tua, yang menggambarkan tingkat kehijauan tanaman padi. Pemberian pupuk N berdasarkan hasil pengukuran warna daun dengan BWD skala 4 dapat menghemat pemakaian pupuk 15-20% dari takaran yang umum digunakan petani tanpa menurunkan hasil [1].

Penelitian tentang deteksi kebutuhan nitrogen pada padi sebelumnya pernah dilakukan pada tahun 2018 dengan menggunakan metode Ekstraksi Ciri Warna RGB, K-Nearest Neighbor (KNN) dan Euclidean Distance [3]. Pada penelitian ini skenario uji coba dilakukan dengan jumlah data gambar sampel daun padi yang diambil melalui kamera, yaitu sebanyak 120 gambar, setiap level warna diuji dengan sepuluh citra daun yang diambil pada pagi, siang dan sore hari. Dari hasil pengujian yang telah dilakukan akurasi yang dihasilkan masih bernilai sedang, yaitu sebesar 66,7%. Penelitian selanjutnya pada tahun 2018 telah dilakukan untuk menentukan tingkat kemanisan buah mangga berdasarkan fitur warna dengan mengimplementasikan metode Support Vector Machine dan K-Nearest Neighbour (K-NN) menghasilkan nilai akurasi yaitu sebesar 87,5% untuk SVM dan 83,3% untuk KNN sehingga akurasi SVM lebih unggul dibanding dengan KNN [4]. Berdasarkan paparan sebelumnya, maka dalam penelitian ini digunakan metode Support Vector Machine (SVM) sebagai klasifikasi dalam mendeteksi kebutuhan unsur nitrogen untuk penentuan jumlah pupuk tanaman padi berdasarkan warna daun.

2. ISI PENELITIAN

2.1 Model Warna HSV

HSV adalah model warna yang lebih baik untuk digunakan untuk berbagai keperluan pengolahan citra dan computer vision. Misalnya saja pada object tracking berdasarkan warna, segmentasi citra dsb. Hue (H) adalah ukuran dari jenis warna seperti warna merah, kuning, hijau dan seterusnya. Representasinya dalam bentuk derajat dengan nilai 0 – 360. Saturasi (S) adalah keberwarnaan suatu warna. Semakin berwarna sebuah warna berarti semakin besar nilai saturasinya. Namun apabila suatu warna pucat, itu berarti saturasinya rendah. Value (V) adalah nilai kecerahan sebuah warna. Warna cerah memiliki nilai Value tinggi dan sebaliknya untuk warna yang gelap [5].

$$V_m = \min R, G, B \quad (1)$$

$$S = \begin{cases} 0 & \text{jika } V = 0 \\ \frac{V_m}{V} & \text{jika } V > 0 \end{cases} \quad (2)$$

$$H = \begin{cases} 0^\circ & \text{jika } S = 0 \\ 60^\circ \times \left(\frac{G-B}{V_m} \bmod 6 \right) & \text{jika } V = R \\ 60^\circ \times \left(2 + \frac{B-R}{V_m} \right) & \text{jika } V = G \\ 60^\circ \times \left(4 + \frac{R-G}{V_m} \right) & \text{jika } V = B \end{cases} \quad (3)$$

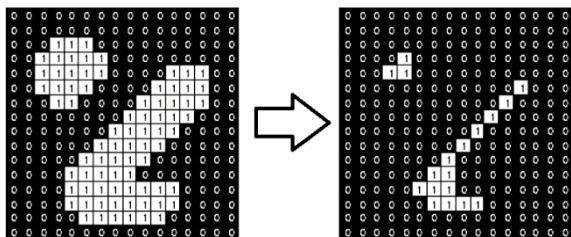
2.2 Erode

Operasi erode (Erosi) mempunyai efek memperkecil struktur citra. Operasi erosi dapat dirumuskan sebagai berikut [6] :

$$A \ominus B = \{ p \in z^2 \mid (a + b) \in |, \text{ untuk setiap } b \in B \} \quad (4)$$

Dimana A merupakan f(x,y) dari citra asli dan B adalah elemen penstruktur atau biasa disebut strel. Elemen penstruktur yang biasa digunakan dalam operasi erosi adalah bentuk kotak. Bentuk elemen penstruktur lainnya ada yang berupa elipse, garis, piringan dan lainnya.

Hasil erosi biasanya merupakan operasi nalar AND dari setiap koordinat A dan B. Berikut merupakan hasil dari operasi dilate terdapat pada Gambar 1.



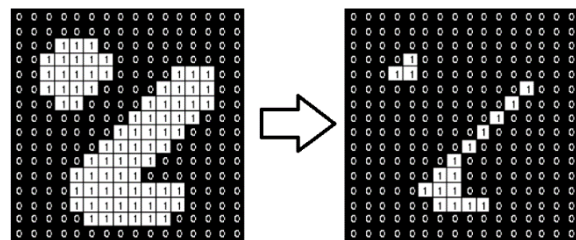
Gambar 1. Hasil Gambar Erode

2.3 Dilate

Operasi dilate (Dilasi) biasa dipakai untuk mendapatkan efek pelebaran terhadap piksel bernilai 1. Operasi dilasi dapat dirumuskan sebagai berikut [6]:

$$A \oplus B = \{ z \mid z = a + b, \text{ dengan } a \in A \text{ dan } b \in B \} \quad (5)$$

Dimana A merupakan f(x,y) dari citra asli dan B adalah elemen penstruktur atau biasa disebut strel. Elemen penstruktur yang biasa digunakan dalam operasi dilasi juga biasanya adalah berbentuk kotak. Hasil dilasi berupa penjumlahan seluruh pasangan koordinat dari A dan B. Berikut merupakan hasil dari operasi dilate terdapat pada Gambar 3.



Gambar 2. Hasil Gambar Dilate

2.4 Histogram Warna

Histogram warna merupakan fitur yang paling banyak digunakan untuk merepresentasikan ciri warna suatu citra. Citra pada umumnya dikonversi ke dalam suatu ruang warna tertentu, kemudian setiap komponen ruang warna dibuat histogramnya. Ruang warna HSV pada umumnya sering digunakan karena ruang warna tersebut dekat dengan persepsi manusia. Namun demikian akan lebih baik pemilihan ruang warna didasari pada objek yang dihadapi [7]. Secara matematis histogram citra dihitung dengan rumus :

$$h_i = \frac{n_i}{n}, \quad i = 0, 1, \dots, L - 1 \quad (6)$$

yang dalam hal ini ,

n_i = jumlah pixel yang memiliki derajat keabuan i
 n = jumlah seluruh pixel di dalam citra

2.5 Intensitas Citra Keseluruhan

Pada sebuah histogram dapat diketahui intensitas citra secara keseluruhan sehingga bisa disimpulkan citra tersebut gelap atau terang. Luminosity (terangnya cahaya) dari sebuah citra dapat diukur dengan menghitung nilai rata-rata (mean) dari histogram sesuai dengan persamaan berikut ..

$$\text{mean} = \frac{\sum i H(i)}{N} \quad (7)$$

Semakin tinggi nilai rata-rata, maka semakin tinggi kecerahan suatu citra

2.6 Kontras Citra

Secara kuantitatif, nilai kontras suatu citra dapat dihitung berdasarkan standar deviasinya dengan menggunakan persamaan

$$\text{var} = \frac{\sum(i - \text{mean})^2 \cdot H(i)}{N} \quad (8)$$

$$\text{std} = \sqrt{\text{var}} \quad (9)$$

Tabel 1. Tabel parameter pengujian algoritma

Parameter	Nilai
Maksimal Perulangan	10, 50, 100
Rasio Pembelajaran	0.1, 0.5, 1
Minimal Error	0.1, 0.01, 0.001

2.7 Open CV

OpenCV (Open Source Computer Vision Library) is a software library intended for image processing that is commonly used in real-time (at that time). This library is made by Intel and is a library that is free to use and is in the shade of the open source (Open source) of the license. This library can also be used on various platforms and is dedicated mostly for real-time image processing [8]. Generally this library uses the C / C ++ programming language, but recently it has been developed in various programming languages such as Python and Java.

2.8 Support Vector Machine

SVM saat pertama kali diperkenalkan oleh Vapnik, hanya dapat mengklasifikasikan data ke dalam dua kelas (klasifikasi biner). Namun, penelitian lebih lanjut untuk mengembangkan SVM sehingga bisa mengklasifikasi data yang memiliki lebih dari dua kelas, terus dilakukan. Ada dua pilihan untuk mengimplementasikan multiclass SVM yaitu dengan menggabungkan beberapa SVM biner atau menggabungkan semua data yang terdiri dari beberapa kelas ke dalam sebuah bentuk permasalahan optimasi. Berikut ini adalah metode yang umum digunakan untuk mengimplementasikan multiclass SVM dengan pendekatan sebagai berikut [9]:

Metode one-against-all

Dengan menggunakan metode ini, akan dibangun k buah model SVM biner (k adalah jumlah kelas). Contohnya, terdapat permasalahan klasifikasi dengan 4 buah kelas. Untuk pelatihan digunakan 4 buah SVM biner seperti pada Tabel 2 dan penggunaanya dalam mengklasifikasi kelas pada data baru dapat dilihat pada persamaan sebagai berikut [10]:

$$\text{Kelas } x = (x_i, x) = \arg \max_{j=1..k} ((w^{(j)})^T \cdot \varphi(x) + b^{(j)}) \quad (10)$$

Tabel 2. Contoh 4 SVM biner dengan metode One-against-all

$y_1 = 1$	$y_1 = -1$	Hipotesis
Kelas 1	Bukan kelas 1	$f^1(x) = (w^1)x + b^1$

Kelas 2	Bukan kelas 2	$f^1(x) = (w^2)x + b^2$
Kelas 3	Bukan kelas 3	$f^1(x) = (w^3)x + b^3$
Kelas 4	Bukan kelas 4	$f^1(x) = (w^4)x + b^4$

2.9 Kernel Trick

Feature space dalam prakteknya biasanya memiliki dimensi yang lebih tinggi dari vektor input (input space). Hal ini mengakibatkan komputasi pada feature space mungkin sangat besar, karena ada kemungkinan feature space dapat memiliki jumlah feature yang tidak terhingga. Selain itu, sulit mengetahui fungsi transformasi yang tepat. Untuk mengatasi masalah ini, pada SVM digunakan "kernel trick". Fungsi kernel yang umum digunakan adalah sebagai berikut [9]:

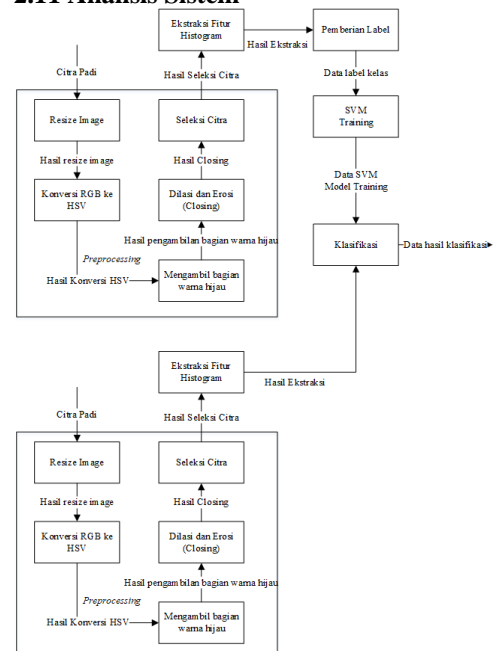
Kernel RBF

$$K(x_i, x) = \exp(-\gamma |x_i - x|^2), \gamma > 0 \quad (11)$$

2.10 Segmentasi Citra HSV

Segmentasi warna berdasarkan deteksi warna HSV merupakan proses segmentasi terkontrol dengan filter warna HSV. Filter warna HSV akan memisahkan warna tertentu sesuai dengan warna acuan dan nilai toleransi tiap elemen warna HSV. Kontrol pengguna melalui sampel warna dan toleransi warna yang menjadi acuan filter sehingga dapat diperoleh segmen dengan warna yang sesuai. Hasil segmentasi warna berdasarkan deteksi warna HSV sangat dipengaruhi oleh sampel warna dan nilai toleransi warna yang menjadi acuan proses segmentasi. Pencahayaan, tekstur dan kontur benda atau latar belakang citra akan sangat mempengaruhi hasil segmentasi dan deteksi objek[10].

2.11 Analisis Sistem

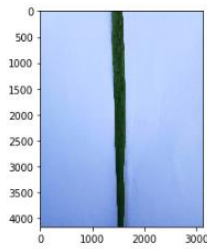


Gambar 3. Deskripsi umum sistem

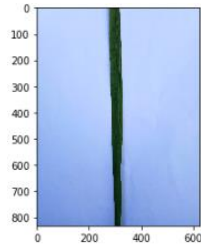
Penjelasan dari tahapan – tahapan diatas adalah sebagai berikut :

1. **Resize Image**

Untuk proses yang lebih cepat dalam mencari parameter pada citra data masukan, citra masukan awal akan di ubah ukurannya dengan proses resize image dari ukuran citra asli menjadi citra ukuran 832 x 624 pixel dengan mode warna RGB (Red,Green, Blue) yang artinya dalam setiap pixel terdapat 3 nilai warna. Berikut adalah gambar citra asli(kiri) dan citra hasil resize(kanan):



Gambar 4. Citra Asli



Gambar 5. Hasil Resize

2. **Konversi RGB ke HSV**

Citra yang sudah di resize sebelumnya merupakan citra RGB (Red,Green,Blue).Proses dalam tahap ini adalah mengubah nilai RGB pada setiap pixel citra menjadi nilai citra HSV Sebagai contoh ambil nilai RGB pada pixel (1,1), maka proses perhitungan konversi RGB ke HSV menggunakan persamaan (1) sebagai berikut :

$$V = \max(184,187,196)$$

$$V = 196$$

$$S = \begin{cases} \frac{V_{\min}(R, G, B)}{V} & \text{jika } V \neq 0 \\ 0 & \text{jika } V = 0 \end{cases}$$

$$S = \frac{V_{\min}(184,187,196)}{196}$$

$$S = 0.06122449$$

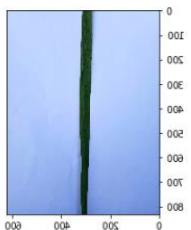
$$H = 240 + 60(18 - 187)$$

$$/(-\min(184,187,196))$$

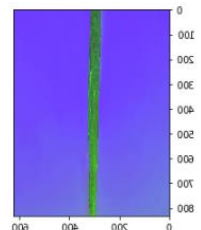
$$H = 225$$

Didapatkan nilai HSV dari pixel (1,1) yaitu H = 225, S = 0.06122449 dan V = 196.

Berikut hasil konversi dari RGB ke HSV



Gambar 6. Citra RGB



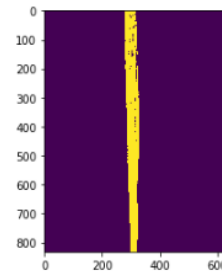
Gambar 7. Citra HSV

3. **Mengambil bagian warna hijau**

Pada proses ini akan dilakukan pengambilan warna HSV hanya pada HSV warna Hijau dan mengabaikan warna lain untuk segmentasi Daun. Proses pengambilan warna hijau HSV ini didasarkan pada skala rentang nilai HSV hijau. Berikut ini merupakan skala rentang nilai warna HSV hijau :

1. Nilai Hue berada dikisaran nilai dari 34 sampai 100
 2. Nilai Saturation dikisaran nilai 100 sampai 255
 3. Nilai Value dikisaran nilai 10 sampai 255
- Berikut ini merupakan flowchart dari menemukan HSV

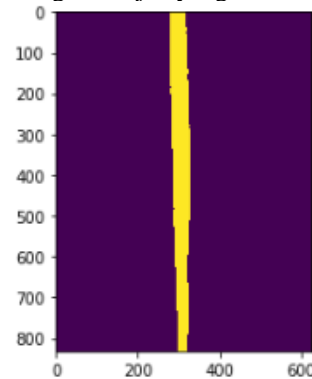
hasil dari pengambilan hanya pada nilai HSV warna hijau terdapat pada Gambar 3.6 sebagai berikut :



Gambar 8. Hasil pengambilan HSV hijau

4. **Dilasi dan Erosi (Closing)**

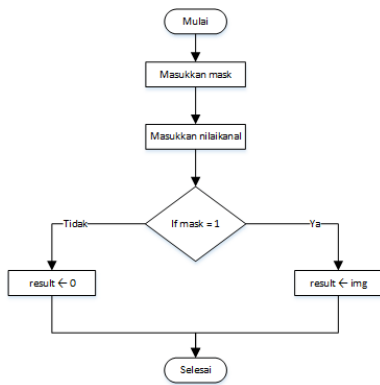
Pada proses ini akan dilakukan closing yaitu melakukan operasi morfologi citra dimana citra akan melalui tahap dilasi terlebih dahulu setelah itu akan dilakukan erosi. Closing bertujuan untuk mengisi lubang kecil pada objek, menggabungkan objek yang berdekatan.



Gambar 9. Hasil Operasi Closing

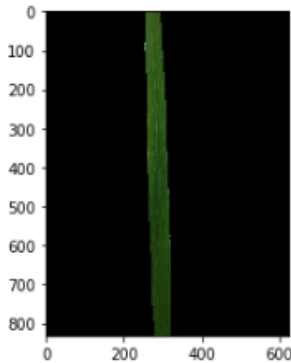
5. **Seleksi Citra**

Setelah mendapatkan pola maka citra Asli akan di seleksi berdasarkan pola hasil dari proses closing sebelumnya. Berikut ini flow chart untuk seleksi citra terdapat pada gambar berikut :



Gambar 10. Flow Chart Seleksi Citra

Hasil dari proses seleksi citra terdapat pada Gambar.11 berikut :



Gambar 11. Citra hasil seleksi

6. Ekstraksi Fitur Histogram

Pada tahapan selanjutnya akan dilakukan ekstraksi fitur dengan menggunakan fitur histogram warna. Metode histogram yang digunakan adalah dengan mengambil nilai luminosity (mean) dan contrast (std) [7]. Dari citra yang sudah diseleksi didapat nilai fitur berikut :

$$\begin{aligned}
 mean_{red} &= 1.7475 & std_{red} &= 7.5822 \\
 mean_{green} &= 3.6057 & std_{green} &= 14.6895 \\
 mean_{blue} &= 1.875 & std_{blue} &= 8.0314
 \end{aligned}$$

7. Pemberian Kelas

Setelah data berhasil mendapatkan nilai fitur kemudian citra yang sudah di dapat nilai fiturnya kemudian diberi label dan kelas. Contoh pemberian label dapat dilihat pada Tabel.3

Tabel 3. Pemberian kelas

Fitur Histogram	Label	Kelas
-----------------	-------	-------

$mean_{red} = 1.7475$ $mean_{green} = 3.6057$ $mean_{blue} = 1.8758$ $std_{red} = 7.5822$ $std_{green} = 14.6895$ $std_{blue} = 8.0314$	+1	Level 2
--	----	---------

8. SVM Training

Untuk melakukan *training* diambil 4 data dan diberi label dan kelas seperti pada Tabel

Tabel 4. Data *training* yang sudah diberi label dan kelas

Fitur Histogram	Label	Kelas
$mean R = 3.274$ $mean G = 6.122$ $mean B = 1.8645$ $std R = 11.7782$ $std G = 21.441$ $std B = 7.6759$	+1	Level 2
$mean R = 1.7475$ $mean G = 3.6057$ $mean B = 1.8758$ $std R = 7.5822$ $std G = 14.6895$ $std B = 8.0314$	-1	Level 3
$mean R = 3.1356$ $mean G = 6.138$ $mean B = 3.3455$ $std R = 10.7392$ $std G = 20.0358$ $std B = 11.9367$	-1	Level 4
$mean R = 1.6592$ $mean G = 4.2026$ $mean B = 4.4214$ $std R = 6.5381$ $std G = 16.1857$ $std B = 16.9892$	-1	Level 5

Selanjutnya dari tiap citra yang sudah di ekstraksi diberi nama x_1, x_2, x_3 dan x_4

Tabel 5. Data x_1, x_2, x_3 dan x_4

Da ta	Mea n R	Mea n G	Mea n B	Std R	Std G	Std B
x ₁	3.274	6.122	1.8645	11.7782	21.441	7.6759
x ₂	1.7475	3.6057	1.8758	7.5822	14.6895	8.0314
x ₃	3.1356	6.138	3.3455	10.7392	20.0358	11.9367
x ₃	1.6592	4.2026	4.4214	6.5381	16.1857	16.9892

Lakukanlah perhitungan $x - x_i$. Hasil dari perhitungan tersebut dapat dilihat pada tabel

Tabel 6. Hasil pengurangan $x - x_i$

Da ta lati h	Mean R	Mean G	Mean B	Std R	Std G	Std B
x ₁ - x ₁	0	0	0	0	0	0
x ₁ - x ₂	1.5265	2.5163	0.0113	4.196	6.7515	0.3555
x ₁ - x ₃	0.1384	-0.016	-1.481	1.039	1.4052	4.2608
x ₁ - x ₄	1.6148	1.9194	2.5569	5.2401	5.2553	9.3133

Selanjutnya lakukan oprasi untuk menghitung panjang vector dengan persamaan sebagai berikut :

$$\|x - x_i\|$$

Untuk Xi maka perhitungannya sebagai berikut :

$$\|x_1 - x_1\| = \sqrt{0.0000^2 + 0.0000 + 0.0000^2 + 0.0000^2 + 0.0000^2 + 0.0000^2} = 0.0000$$

Maka didapatlah hasil untuk $x_i = 0.0000$, lakukanlah perhitungan tersebut untuk nilai x_i selanjutnya

Berikutnya panjang vektor kemudian dimasukan ke dalam persamaan $\exp(-\gamma \|x - x_i\|^2)$ sebagai berikut:

$$K(1,1) = \exp(-\gamma \|x_1 - x_1\|^2) = \exp(-0,1(0.0000^2)) = \exp(0) = 1$$

Lakukan perhitungan tersebut untuk nilai x_i selanjutnya. Sehingga hasil perhitungan Nilai K Dengan Kernel RBF sepeti pada Tabel 7

Tabel 7. Hasil Perhitungan Nilai K dengan kernel RBF

K(1,1)	K(1,2)	K(1,3)	K(1,4)
1	0.000748	0.096123	1.92E-07

K(2,1)	K(2,2)	K(2,3)	K(2,4)
0.000748	1	0.001612	0.000118
K(3,1)	K(3,2)	K(3,3)	K(3,4)
0.096123	0.001612	1	0.001491
K(4,1)	K(4,2)	K(4,3)	K(4,4)
1.92E-07	0.000118	0.001491	1

Kemudian tahap selanjutnya adalah melakukan perhitungan terhadap y . Untuk nilai y adalah nilai dari label. Nilai y tersebut dapat dilihat pada Tabel 8.

Tabel 8. Nilai label pada y

y1	y2	y3	y4
1	-1	-1	-1

Selanjutnya dilakukan perhitungan $y_i y_j$ sebanyak N data. Hasil dari perhitungan tersebut dapat dilihat pada Tabel 9

Tabel 9. Perhitungan $y_i y_j$

y1 y1	y1 y2	y1 y3	y1 y4
1	-1	-1	-1
y2 y1	y2 y2	y2 y3	y2 y4
-1	1	1	1
y3 y1	y3 y2	y3 y3	y3 y4
-1	1	1	1
y4 y3	y4 y3	y4 y3	y4 y4
-1	1	1	1

Dengan menggunakan kernel K sebagai pengganti dot-product $x_i x_j$ dalam persamaan dualitas Lagrange multiplier didapatkan

$$Ld \max = (a_1 + a_2 + a_3 + a_4) - \frac{1}{2} (a_1^2 - 0.001496513 a_1 a_2 - 0.19224543 a_1 a_3 - (3.84534E - 07 a_1 a_4) + a_2^2 + 0.003224797 a_2 a_3 + 0.000236784 a_2 a_4 + a_3^2 + 0.002981918 a_3 a_4 - a_4^2)$$

$$\text{Syarat 1 : } a_1 - a_2 - a_3 - a_4 = 0$$

$$\text{Syarat 2 : } a_1, a_2, a_3, a_4 \geq 0$$

Dalam fungsi tujuan, suku kedua sudah dikalikan dengan $y_i y_j$. Persamaan tersebut memenuhi standar *Quadratic programming* sehingga bisa dibantu penyelesaiannya dengan solver komersial untuk

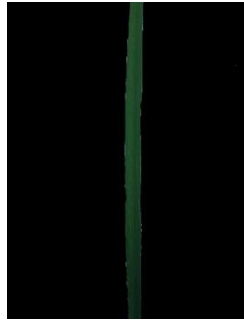
Quadratic Programming(QP). Dengan bantuan perangkat lunak, didapatkan hasil sebagai berikut.

- $a_1 = 1.499994078$
- $a_2 = 0.499998026$
- $a_3 = 0.49998982$
- $a_4 = 0.499998026$
- $b = -0.499985873$

9. Klasifikasi (SVM Testing)

Setelah berhasil melakukan perhitungan Training selanjutnya dilakukan perhitungan SVM Testing contoh data yang dijadikan sampel seperti pada Tabel 10.

Tabel 10. Tabel Data Testing

Citra data testing	Fitur Histogram
	Mean R = 2.0813 Mean G = 4.5282 Mean B = 2.2206 Std R = 7.9187 Std G = 16.4213 Std B = 8.7611

Lakukanlah perhitungan $x_{trainingi} - x_{testing}$. Hasil dari perhitungan tersebut dapat dilihat pada tabel

Tabel 11. Tabel Data Testing

Data	Mea n R	Mea n G	Mea n B	Std R	Std G	Std B
$x_{training1} - x_{testing}$	1.1927	1.5938	0.3561	3.8595	5.0197	1.0852
$x_{training2} - x_{testing}$	0.3338	0.9225	0.3448	0.3365	1.7318	0.7297
$x_{training3} - x_{testing}$	1.0543	1.6098	1.1249	2.8205	3.6145	3.1756
$x_{training4} - x_{testing}$	0.4221	0.3256	2.2008	1.3806	0.2356	8.2281

Selanjutnya lakukan perhitungan untuk vektor

$$\|x_{trainingi} - x_{testing}\|$$

Sehingga didapat nilai Panjang vektor seperti pada Tabel 11

Tabel 11. Tabel Hasil Panjang Vektor

$\ x_{training1} - x_{testing}\ $	6.7350
$\ x_{training2} - x_{testing}\ $	2.1739

$\ x_{training3} - x_{testing}\ $	6.006052814
$\ x_{training4} - x_{testing}\ $	8.648173087

$$K(x_{training1}, x_{testing}) = \exp(-\gamma \|x_{training1} - x_{testing}\|^2) = \exp(-0,5(6.735007522)^2) = 0.010715836$$

Lakukan perhitungan tersebut untuk $x_{training} 2,3$ dan 4

$$f(x_{testinglevel2}) = (((1.499994078 * 1 * (0.010715836)) + (0.499998026 * (0.49998982 * (-1) * (0.02712588) + (0.499985873 * (-0.809441282))))$$

$$\max(f(x_{testing})) = [(-0.809441282)(0.443548388) (-0.152737512)(-0.179254946)]$$

Berdasarkan nilai maksimum $f(x_{testing})$, maka hasil dari klasifikasi multiclass adalah bahwa data uji merupakan bagian dari kelas level 3.

2.7 Pengujian Confusion Matrix

Pengujian akurasi menggunakan confusion matrix dilakukan untuk menguji performansi dari aplikasi deteksi kebutuhan nitrogen padi. Pengujian akurasi tersebut dibuat menjadi tiga waktu sesuai pengambilan citra yang dijadikan dataset yaitu pagi,siang dan sore.

Skenario 1 : 60 data training dan 60 data testing

Tabel 12. Hasil Akurasi Skenario 1

Akurasi					
Waktu	Level 2	Level 3	Level 4	Level 5	Rata-rata
Pagi	85%	60%	60%	75%	83%
Siang	55%	70%	75%	100%	70%
Sore	80%	65%	67%	92%	75%
Semua	100%	65%	65%	100%	76%

Skenario 2 : 72 data training dan 48 data testing

Tabel 13. Hasil Akurasi Skenario 2

Akurasi					
Waktu	Level 2	Level 3	Level 4	Level 5	Rata-rata
Pagi	100%	75%	75%	100%	88%
Siang	81%	69%	63%	75%	72%
Sore	50%	75%	69%	94%	72%
Semua	77%	73%	69%	90%	77%

Skenario 3 : 84 data *training* dan 36 data *testing*

Tabel 13. Hasil Akurasi Skenario

Akurasi					
Waktu	Level 2	Level 3	Level 4	Level 5	Rata-rata
Pagi	100%	75%	75%	100%	88%
Siang	75%	83%	58%	83%	75%
Sore	50%	75%	69%	94%	72%
Semua	75%	81%	67%	94%	79%

Dari 3 Skenario pengujian tersebut diketahui bahwa akurasi tertinggi keseluruhan yaitu pada pengujian scenario 3 dengan menggunakan 84 data *training* dan 36 data *testing* dengan nilai akurasi sebesar 79% sementara dari 3 skenario pengujian tersebut citra yang diambil pada pagi hari selalu memiliki nilai akurasi tertinggi yaitu sebesar 88% hal ini dikarenakan intensitas cahaya pada pagi hari tidak terlalu tinggi sehingga warna yang dihasilkan lebih jelas.

3. PENUTUP

3.1 Kesimpulan

Berdasarkan dari hasil pengujian yang dilakukan, dapat diambil kesimpulan sebagai berikut. Secara keseluruhan akurasi terbesar yang diperoleh dari implementasi SVM pada deteksi kebutuhan nitrogen padi adalah 79% sedangkan berdasarkan waktu pengambilan gambar yang terbaik adalah pada pagi hari dengan akurasi yang didapat adalah sebesar 88%. Faktor yang mempengaruhi dari hal tersebut adalah cahaya. Pada pagi instensitas cahaya tidak terlalu tinggi sehingga warna yang dihasilkan terlihat lebih jelas.

3.2 Saran

Disamping kelebihan – kelebihan suatu sistem pasti akan mempunyai kekurangan, maka dari itu pengembangan lebih lanjut tentang penelitian ini merupakan saran – saran yang dapat diberikan sebagai berikut :

1. Dilakukan penambahan data dengan akuisisi citra yang lebih variatif, baik dari ukuran, resolusi citra, jenis kamera, dan aspek teknis akuisisi lainnya sehingga data yang digunakan dapat menangani data uji yang bervariasi.
2. Untuk memperoleh data latih yang lebih presisi, sebaiknya pada saat akuisisi data citra selalu didampingi oleh peneliti yang berkompeten serta perhatikan kualitas daun padi yang akan dipilih.

DAFTAR PUSTAKA

- [1] Erythrina, “Leaf Color Chart: a Tool to Increase Nitrogen,” *J. Litbang Pert*, vol. 35, pp. 1–10, 2016.
- [2] F. O. Oroka, “Responses of Rice and Cowpea Intercropping to Nitrogen Fertilizer and Plant Population (2): Vegetative Growth and Correlates of Yield and Yield Components,” vol. 2, no. 4, pp. 174–179, 2011.
- [3] E. B. Setiawan and R. Herdianto, “Penggunaan Smartphone Android sebagai Alat Analisis Kebutuhan Kandungan Nitrogen pada Tanaman Padi,” *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 7, no. 3, 2018.
- [4] M. Ichwan, I. A. Dewi, and Z. M. S, “Klasifikasi Support Vector Machine (SVM) Untuk Menentukan TingkatKemanisan Mangga Berdasarkan Fitur Warna,” vol. 3, no. 2, pp. 16–23, 2018.
- [5] R. Y. Prabowo, Rahmadwati, and P. Mudjirahardjo, “Klasifikasi Kandungan Nitrogen berdasarkan Warna Daun melalui Color Clustering menggunakan Metode Fuzzy C Means dan,” *J. EECCIS*, vol. 12, no. 1, pp. 1–8, 2018.
- [6] H. Priyanto, *Pengolahan Citra Digital Teori dan Aplikasi Nyata*. Bandung: Informatika Bandung, 2017.
- [7] K. Abdul and S. Kodir, *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Penerbit Andi, 2013.
- [8] A. Yulio, “Mengenal OpenCV (Open Source Computer Vision Library),” 2017. [Online]. Available: <https://devtrik.com/opencv/mengenal-opencv-open-source-computer-vision-library/>. [Accessed: 21-Feb-2017].
- [9] E. Prasetyo, *DATA MINING Mengolah Data Menjadi Informasi Menggunakan Matlab*. Yogyakarta: Penerbit Andi, 2015.
- [10] B. Y. Budi Putranto, W. Hapsari, and K. Wijana, “Segmentasi Warna Citra Dengan Deteksi Warna Hsv Untuk Mendeteksi Objek,” *J. Inform.*, vol. 6, no. 2, 2011.