# Aplikasi Pendeteksi Kerusakan Pada Daun Berdasarkan Warna

**M Ridwan Dwi Septian[1)], Margi Cahyanti[2)], Ericks Rachmat Swedia[3)]**
Teknik Informatika, Fakultas Teknologi Informasi, Universitas Gunadarma
Jalan Margonda Raya No.100, Pondok Cina, Depok
e-mail: ridwandwiseptian@staff.gunadarma.ac.id [1)], margi@staff.gunadarma.ac.id [2)],
ericks_rs@staff.gunadarma.ac.id [3)]

***Abstrak***

*Penelitian ini membuat sebuah aplikasi yang akan mendeteksi kerusakan pada citra daun dengan memberikan persentase berapa kerusakan pada daun yang di ujikan dengan metode pengolahan citra. Proses mendeteksi kerusakan pada daun dilakukan dengan metode pengolahan citra terdiri dari tahapan transformasi ke HSL dengan mengambil range warna daun yang tidak rusak. Selanjutnya citra daun yang awal diubah menjadi citra biner agar mendapatkan keseluruhan bentuk daun. Tahapan selanjutnya untuk mendapatkan persentase kerusakan dengan cara seluruh jumlah piksel pada citra biner dikurang dengan jumlah piksel citra yang dirubah ke HSL, sehingga bisa mendapatkan kerusakan piksel pada daun. Hasil pengujian pada kerusakan daun ini dapat menampilkan seluruh output berupa persentase keruskan daun dari 10 data citra daun mangga. Uji coba aplikasi ini menggunakan 10 data citra daun mangga. Citra ini diambil berukuran 100 x 56 piksel. Dengan adanya aplikasi ini dapat mempermudah user dalam memperkirakan kerusakan citra daun.*

***Kata kunci:*** *biner, citra, daun, HSL, mangga, warna.*

## 1. Pendahuluan

Sejak ditemukannya komputer pertama kali, manusia terus melakukan penelitian untuk menciptakan cara – cara dalam berinteraksi dengan dunia maya yang diciptakan komputer tersebut. Dimulai dari tampilan interaksi berbasis teks (*text based interface*) yang masih terbatas pada command line yang digunakan pada komputer generasi pertama sejak tahun 1940, hingga ditemukannya *graphic user interface* (GUI) yang biasa kita gunakan saat ini. Salah satu bidang ilmu dan teknologi komputer yang berkembang adalah pengolahan citra [1].

Saat ini tanaman banyak mengalami kerusakan. Serangga adalah salah satu penyebab utama dalam kerusakan tanaman. Hal ini juga merusak pada rantai makanan. Para ilmuwan tanaman biasanya memperkirakan kerusakan daun dengan menghitung persentase daerah daun yang mengalami kerusakan [2].

Pohon mangga berperawakan besar, dapat mencapai tinggi 40 meter atau lebih, meski kebanyakan mangga peliharaan hanya sekitar 10 meter atau kurang. Batang mangga tegak, bercabang agak kuat dengan daun-daun lebat membentuk tajuk yang indah berbentuk kubah, oval atau memanjang, dengan diameter sampai 10 meter. Kulit batangnya tebal dan kasar dengan banyak celahcelah kecil dan sisik-sisik bekas tangkai daun. Warna pepagan (kulit batang) yang sudah tua biasanya coklat keabuan, kelabu tua sampai hampir hitam [9].

Salah satu permasalahan yang muncul adalah sulitnya menghitung persentase daerah daun yang mengalami kerusakan secara manual. Berdasarkan permasalahan diatas, penelitian ini mencoba mendeteksi kerusakan pada daun yang diimplementasikan dalam bentuk aplikasi dengan menggunakan pengolahan citra sehingga dapat diketahui dengan cepat apakah dalam selembar daun terdapat kerusakan atau tidak dan menghitung persentase daerah daun yang mengalami kerusakan.

## 2. Metode Penelitian

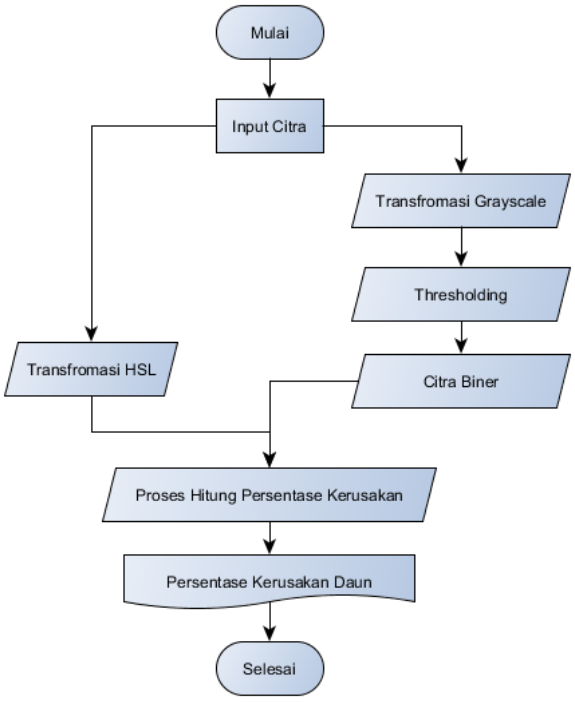
Dalam penelitian ini, metode penelitian yang dilakukan oleh peneliti melalui beberapa tahapan, diantaranya :

1. Pengumpulan Data
   - Citra daun mangga diambil di halaman Universitas Gunadarma.
   - Citra difoto menggunakan smartphone iPhone 6.

- Ukuran piksel citra awal dirubah menjadi 100x56 piksel.
- Citra yang diambil menggunakan *background* berwarna putih.
- Posisi pengambilan citra tegak lurus terhadap lebar daun.

2. Merancang Output dan Implementasi

Disini peneliti merancang tampilan aplikasi yang berisi sebuah persentase kerusakan pada daun dari keseluruhan pada daun tersebut beserta dengan piksel-pikselnya, aplikasi ini menggunakan bahasa pemograman C#.




Gambar 1. Alur Program

Keterangan alur:

- Diawali dengan menerima *input* citra berupa file sampel yang sudah ditentukan sebelumnya.
- Citra *input* ditransformasikan ke bentuk HSL.
- Citra *input* ditransformasikan ke bentuk *grayscale*, setelah citra menjadi *grayscale* citra tersebut diambil ambang batas menggunakan *thresholding* untuk menjadikan citra tersebut menjadi citra biner.
- Proses Persentase Hitung Keseluruhan, pada tahapan ini citra yang ditransformasi RGB ke HSL dan transfromasi *thresholding* dihitung jumlah pikselnya masing-masing dan mendapatkan hasil.
- Menampilkan nilai persentase kerusakan.

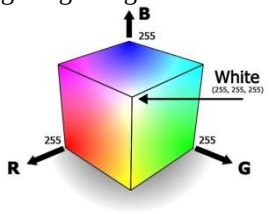
### 2.1. Citra

Sebuah citra didefinisikan sebagai gambar yang terdapat pada bidang dua dimensi. Secara matematis citra merupakan fungsi menerus (kontinu) dari intensitas cahaya pada bidang dua dimensi [5]. Suatu citra digital terbentuk dari kumpulan titik pembentuk citra yang disebut dengan piksel. Piksel merupakan data yang mengandung intensitas cahaya yang dinyatakan dalam bilangan bulat [4], sehingga citra digital adalah kumpulan piksel yang dinotasikan dalam bentuk bilangan pada sejumlah baris dan kolom.

### 2.2. Model Warna RGB

Dalam model warna RGB, setiap warna memperlihatkan komponen spektrum *red*, *green*, dan *blue*. Model ini didasarkan pada sistem koordinat kartesian. *Sub-space* warna yang dicari adalah kubus

dimana nilai RGB pada tiga sudut; *cyan*, *magenta*, dan *yellow* ada pada tiga sudut lain hitam adalah *origin* dan putih adalah titik paling jauh dari *origin*. Dalam model ini, *grayscale* (titik-titik nilai *equal* RGB) diperluas dari hitam ke putih, sepanjang garis gabungan dua titik. [3].




Gambar 2. Model Warna RGB

### 2.3. Model Warna HSL

Model HSL merupakan metode yang ditemukan oleh Alvy Ray Smith pada tahun 1978. Model ini mempresentasikan warna dalam tiga komponen: *hue*, *saturation*, dan *lightness* [7].

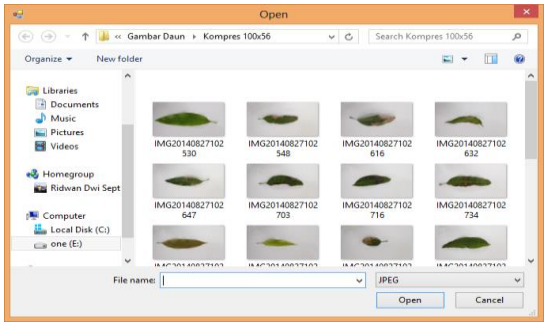
- *Hue* adalah warna yang direfleksikan ataupun ditransmisikan sebuah objek. Nilainya diukur dari lokasi pada roda standar warna, yang diekspresikan dengan nilai derajat sudut di antara 0° dan 360°. Dalam penggunaannya, *hue* mengidentifikasikan nama dari sebuah warna seperti merah, *orange* (jingga), atau hijau.
- *Saturation*, sering dikenal dengan *chroma*, yaitu ukuran atau kemurnian sebuah warna, *saturation* merepresentasikan ukuran (kuantitas) dari proporsi keabuan pada *hue*, ukurannya dalam bentuk persentase dari 0% (*gray*) sampai dengan 100% (*fully saturated*). Pada roda standar warna, nilai saturation dari pusat roda (lingkaran) menuju tepian roda akan semakin bertambah.
- *Lightness* adalah sebuah ukuran *relative* skala pencahayaan (*lightness*) atau kegelapan (*darkness*) dari sebuah warna, umumnya diukur sebagai persentase dari 0% (*black*) sampai dengan 100% (*white*).

### 2.4. Citra Biner

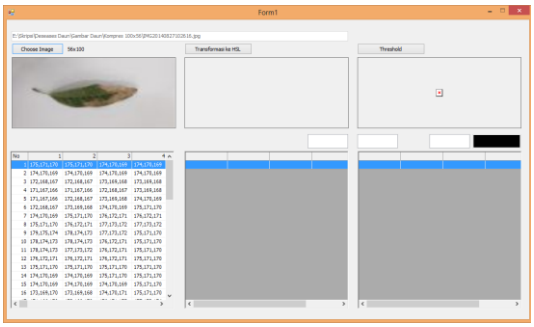
Citra biner adalah bentuk citra direprentasikan berupa angka biner yaitu 0 dan 1. Untuk dapat menghasilkan citra biner diperlukan data gambar *grayscale*, selanjutnya *thresholding* dapat digunakan untuk membuat citra biner [3]. *Thresholding* adalah metode untuk merubah *gray scale* image menjadi *binary image* sehingga objek yang diinginkan terpisah dari latar belakangnya[8]. *Thresholding* merupakan metode paling sederhana, dimana tiap objek atau *region image* dibedakan berdasarkan penyerapan cahaya atau reflektifitas konstan pada permukaannya. Suatu nilai *threshold* (nilai konstan *brightness*) dapat ditentukan untuk membedakan objek dengan latar belakangnya. Tujuan dari *thresholding* adalah untuk memisahkan piksel yang mempunyai nilai keabuan (*gray value*) lebih tinggi dengan yang lebih rendah, misalnya piksel yang nilai keabuannya lebih tinggi diberi nilai biner 1 sedangkan piksel dengan nilai keabuan lebih rendah diberi nilai biner 0.

## 3. Hasil dan Pembahasan

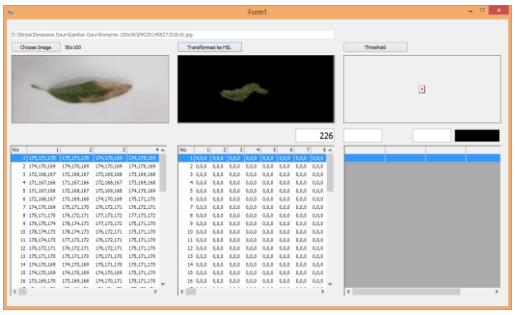
Tahap ini adalah tahapan terakhir untuk memastikan bahwa aplikasi yang telah dibuat dapat dioperasikan dengan baik, di bawah ini merupakan implementasi hasil uji coba aplikasi :



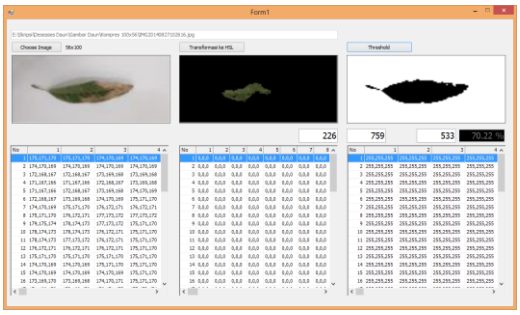

Gambar 3. Pemilihan Gambar




Gambar 4. Hasil *input* gambar



Gambar 5. Citra dirubah ke HSL



Gambar 6. Hasil *threshold*

Hasil dari *output*nya berupa persentase kerusakan yang diambil dari :

- Citra yang di*input* seperti gambar di bawah ini :

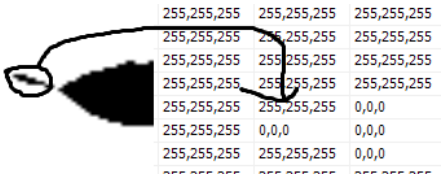
Gambar 7. Citra *Input*

- Citra *input* tersebut ditransformasi kebentuk HSL dengan filter :

Dengan mengambil nilai ambang dari *Hue* = (65, 140), *Saturation* = (0.2f, 1), dan *Luminance* = (0.1f, 1) dapat mempresentasikan citra yang akan diambil, dengan mengambil nilai dari HSL akan menampilkan citra yang lebih baik dari pada RGB, karena dapat melihat bentuk warna yang menghasilkan citra seperti gambar di bawah ini :

Gambar 8. Transformasi RGB ke HSL

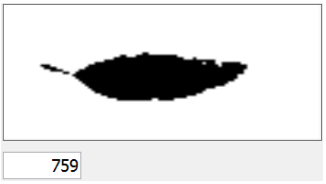
- Citra *input* ditransformasi kedalam citra biner dengan langkah awal mengubah citra tersebut ke *grayscale*, lalu citra *grayscale* tersebut di *threshold* dengan nilai 150. Jika nilai dari *grayscale* lebih dari 150 dianggap 1 dan jika dibawah 150 dianggap 0 seperti gambar di bawah ini:



| | | |
|---|---|---|
| 255,255,255 | 255,255,255 | 255,255,255 |
| 255,255,255 | 255,255,255 | 255,255,255 |
| 255,255,255 | 255,255,255 | 255,255,255 |
| 255,255,255 | 255,255,255 | 255,255,255 |
| 255,255,255 | 255,255,255 | 0,0,0 |
| 255,255,255 | 0,0,0 | 0,0,0 |
| 255,255,255 | 255,255,255 | 0,0,0 |

Gambar 9. Piksel Transformasi RGB ke Cita Biner

- Setelah proses HSL dan binerisasi citra, langkah selanjutnya menghitung persentase keseluruhan daun dan kerusakan pada daun. Dari hasil keseluruhan bentuk daun citra biner didapatkan 759 piksel berdasarkan baris dan kolom pada data tabel seperti gambar di bawah ini.




Gambar 10. Piksel Citra Biner

- Tahapan selanjutnya adalah mendapatkan jumlah piksel dari citra HSL, seperti gambar di bawah ini:


Gambar 11. Piksel Citra HSL

- Untuk mendapatkan persentase kerusakan, digunakan rumus :

$$persentase\ kerusakan = \frac{y - x}{y}\ x\ 100\%$$

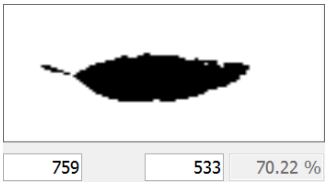
$$persentase\ kerusakan = \frac{759 - 226}{759}\ x\ 100\%$$

$$persentase\ kerusakan = \frac{533}{759}\ x\ 100\%$$

$$persentase\ kerusakan = 70.22\%$$

$dimana$ :
$x = jumlah\ piksel\ citra\ HSL$
$y = jumlah\ piksel\ citra\ biner$

Hasil persentase kerusakan diperlihatkan dalam gambar 12.




Gambar 12. Hasil Persentase Kerusakan Pada Citra Daun

Dari hasil persentase di atas, dapat diketahui persentase kerusakan pada daun tersebut adalah 70,22%.

Pada tahap uji coba, program menggunakan 10 citra daun mangga dengan bentuk dan pola yang berbeda-beda. Tahap uji coba dilakukan dengan memasukkan citra sampel sebagai *input* pada aplikasi yang telah dibuat ditransformasi ke HSL dengan mengambil range warna daun yang tidak rusak. Hasil uji coba aplikasi dapat dilihat pada tabel berikut :

Tabel 1. Hasil Uji Coba

| No | *Input* | HSL | Biner | Persentase Kerusakan % |
|---|---|---|---|---|
| 1 |  |  |  | 3,15% |
| 2 |  |  |  | 53,32% |

| | | | | |
|---|---|---|---|---|
| 3 |  |  |  | 70,22% |
| 4 |  |  |  | 46,99% |
| 5 |  |  |  | 43,65% |
| 6 |  |  |  | 40,75% |
| 7 |  |  |  | 47,47% |
| 8 |  |  |  | 39,46% |
| 9 |  |  |  | 100% |
| 10 |  |  |  | 56% |

**4. Simpulan**

Proses mendeteksi kerusakan pada daun dilakukan dengan tahapan transformasi ke citra HSL dengan mengambil *range* warna daun yang tidak rusak. Selanjutnya citra yang di*input* diawal tadi diubah menjadi citra biner dengan *threshold* sebesar 150, agar mendapatkan keseluruhan bentuk daun. Untuk mendapatkan persentase kerusakan dilakukan dengan cara menghitung jumlah piksel pada citra biner dikurangi dengan jumlah piksel citra HSL.

Uji coba menggunakan 10 data citra daun mangga. Citra uji coba diambil berukuran 100 x 56 piksel. Aplikasi telah berhasil mengeluarkan output berupa persentase kerusakan pada daun.

Aplikasi ini hanya menghitung kerusakan daun berdasarkan warna daun. Untuk pengembangan selanjutnya, sebaiknya ditambahkan analisa kerusakan berdasarkan tekstur dan juga dapat menentukan jenis penyakit pada daun.

**Daftar Pustaka**

[1] Witarto. Memahami Sistem Informasi. Bandung : Informatika Bandung. 2004.
[2] Jayamala K.Patil, Raj Kumar. Advance In Image Processing For Detection of Plant Diseases. Pune : Deemed University. 2011.
[3] Shapiro, Linda G., Stockman, George C. Computer Vision. 2002.
[4] Ahmad, U. Pengolahan Citra Digital Dan Teknik Pemrogramannya. Yogyakarta : Graha Ilmu. 2005.
[5] Rinaldi Munir. Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung : Informatika Bandung. 2004.
[6] Prasetyo, Eko. Pengolahan Citra Digital dan Aplikasinya Menggunakan Matlab. Yogyakarta : Andi. 2011.
[7] Swedia ER., Cahyanti Margi. Algoritma Transformasi Ruang Warna. Depok: Indie Publishing. 2015.
[8] Ramesh Jain, Rangachar Kasturi, Brian G. Schunck Published by McGraw-Hill, *Inc.* Machine vision. Singapore : McGraw-Hill. 1995.
[9] Verheij, E.W.M. dan R.E. Coronel. Sumber Daya Nabati Asia Tenggara 2: Buah-buahan yang dapat dimakan. Jakarta : Gramedia. 1997.