

BAB II

TIJAUAN PUSTAKA

A. Landasan Teori

1. Bawang Merah

Bawang merah (*Allium ascalonicum L.*) merupakan komoditi hortikultura yang tergolong sayuran rempah. Bawang merah merupakan salah satu sayuran umbu multi guna. Paling penting digunakan sebagai bahan bumbu dapur sehari-hari dan penyedap makanan. Kegunaan lain dari bawang merah adalah sebagai obat tradisional untuk pelayanan kesehatan masyarakat. Dan fungsi dalam tubuh antara lain adalah memperbaiki dan memudahkan pencernaan serta menghilangkan lendir di tenggorokan (Tumpang et al., n.d.)

2. Pengertian Citra

Citra adalah sebuah representasi visual dalam bentuk dua dimensi yang memuat informasi gambar. Di antara berbagai jenis citra yang ada, citra digital menjadi salah satu yang paling umum digunakan. Citra digital merupakan hasil tangkapan atau perekaman gambar oleh perangkat atau mesin dengan menggunakan pendekatan sampling dan kuantisasi. Pendekatan sampling ini melibatkan pengambilan sejumlah titik dalam gambar asli untuk merepresentasikan informasi visualnya secara diskrit. Citra digital dapat dianggap sebagai matriks, di mana indeks baris dan kolom mewakili titik pada citra, dan elemen matriks menyatakan tingkat keabuan pada titik tersebut. Pemodelan citra digital dalam bentuk matriks berukuran $N \times M$:

Nilai yang berada di baris dan kolom (pada x, y) disebut dengan picture element atau pixels. Jumlah piksel per-satuan panjang akan menentukan resolusi citra tersebut (Hamdani et al., 2017).

0	134	145	231
0	167	201	197
...
0	134	145	231
0	167	201	197

Gambar 2. 1Contoh Matriks Citra

3. Greyscale

Citra grayscale merupakan citra yang hanya memiliki satu nilai kanal di setiap piksel nya, dengan nilai yang menunjukkan tingkat intensitas. Nilai tersebut berkisar dari warna hitam, keabuan hingga putih, dan dapat diperoleh dengan melakukan konversi terhadap citra berwarna menggunakan koefisien tertentu (Wijaya dkk., 2021)

4. Segmentasi Citra

Segmentasi adalah proses membagi citra menjadi beberapa bagian berdasarkan atribut yang serupa. Tujuan segmentasi adalah untuk memisahkan bagian-bagian yang berbeda dalam citra. Tingkat segmentasi ditentukan oleh masalah yang sedang dipecahkan. Jika obyek yang ingin diidentifikasi dalam aplikasi telah terisolasi, maka proses segmentasi dihentikan. Analisis dilakukan untuk mendeteksi adanya penyimpangan tertentu, seperti komponen yang salah atau koneksi yang terputus, pada rakitan produk elektronik yang diinginkan dengan menggunakan pemeriksaan otomatis. Untuk kategori yang pertama, pendekatannya adalah memilah citra yang didasari oleh perubahan kasar dalam intensitas, seperti tepi dalam citra. Pendekatan utama kategori kedua didasarkan pada pemecahan citra ke dalam region yang sama menurut sejumlah kriteria yang didefinisikan, seperti thresholding, region growing, region splitting dan merging (Prasetyo, 2011).

5. Segmentasi K-Means Clustering

Pada proses manualisasi Pre-processing citra, terdapat dua tahap yaitu Rescaling dan perubahan ruang warna RGB menjadi Lab*. Perubahan ruang warna RGB menjadi Lab* melibatkan dua tahap yaitu mengubah ruang warna RGB ke XYZ, kemudian dari XYZ mengubahnya menjadi ruang warna Lab*. Tahap pertama dalam proses Pre-processing adalah memisahkan citra input menjadi setiap komponen variabel dalam ruang warna RGB, yaitu R (red), G (green), dan B (blue). Setiap komponen variabel akan diolah secara terpisah sebelum dilakukan konversi ke ruang warna Lab*..

Manualisasi RGB to L*a*b*

Proses perubahan ruang warna RGB menjadi Lab* melibatkan beberapa tahap. Tahap pertama adalah mengubah ruang warna RGB menjadi XYZ dengan menggunakan transformasi matrix 3x3 dan nilai tristimulus sesuai dengan persamaan(1). Setelah tahap ini selesai, citra akan siap untuk diubah ruang warnanya dari XYZ menjadi Lab* melalui tahap selanjutnya. Dalam konversi ruang warna RGB ke XYZ, transformasi matrix 3x3 yang digunakan adalah sebagai berikut:

$$\begin{aligned}
 [X] &= (0,412453 \times R) + (0,357580 \times G) + (0,180423 \times B) \\
 [Y] &= (0,212671 \times R) + (0,715160 \times G) + (0,072169 \times B) \\
 [Z] &= (0,019334 \times R) + (0,119193 \times G) + (0,950227 \times B)
 \end{aligned}
 \tag{1}$$

Dimana:

- 1) [X], [Y], dan [Z] adalah nilai tristimulus pada ruang warna XYZ
- 2) R, G, dan B adalah nilai tristimulus pada ruang warna RGB

Setelah mengkonversi ruang warna RGB ke XYZ, selanjutnya dilakukan proses pembagian dengan *White Reference*-nya untuk mendapatkan nilai ruang warna Lab*. Penyusun ruang warna Lab* menggunakan persamaan(2), di mana setiap nilai tristimulus XYZ dibagi dengan *White Reference*-nya masing-masing. *White Reference*

merupakan nilai maksimal dari setiap tristimulus pada ruang warna XYZ, yang ditandai dengan X_0 , Y_0 , dan Z_0 . Nilai X_0 adalah 242,36628, Y_0 adalah 255, dan Z_0 adalah 277,63227, sehingga rentang nilai yang dihasilkan dari proses pembagian tersebut berkisar antara 0 sampai 1. Berikut merupakan contoh perhitungan piksel (i, j) pembagian ruang warna dengan *White Reference*-nya:

$$\begin{aligned}\frac{X}{X_0} &= \frac{p_{(i,j)}}{242,3662} \\ \frac{Y}{Y_0} &= \frac{p_{(i,j)}}{255} \\ \frac{Z}{Z_0} &= \frac{p_{(i,j)}}{277,63227}\end{aligned}\tag{2}$$

Pada tahap selanjutnya, dilakukan perhitungan fungsi yang akan menjadi komponen penyusun dari persamaan(9), yaitu dengan simbol $f(X/X_0)$, $f(Y/Y_0)$, dan $f(Z/Z_0)$. Persamaan ini akan digunakan untuk menghitung nilai L^* , a^* , dan b^* dalam ruang warna Lab* berdasarkan nilai XYZ dan *White Reference*-nya, persamaan (9) memiliki dua kondisi, yaitu:

1. Jika hasil pembagian nilai XYZ dengan *White Reference*-nya (X/X_0 , Y/Y_0 , atau Z/Z_0) lebih besar dari 0.008856:

$$f(X/X_0) = X/X_0^{(1/3)}$$

$$f(Y/Y_0) = Y/Y_0^{(1/3)}$$

$$f(Z/Z_0) = Z/Z_0^{(1/3)}$$

2. Jika hasil pembagian nilai XYZ dengan *White Reference*-nya (X/X_0 , Y/Y_0 , atau Z/Z_0) memiliki rentang nilai antara 0 hingga 0.008856:

$$f(X/X_0) = (903.3 * X)/X_0 + 16)/116$$

$$f(Y/Y_0) = (903.3 * Y)/Y_0 + 16)/116$$

$$f(Z/Z_0) = (903.3 * Z)/Z_0 + 16)/116$$

Dengan menggunakan nilai fungsinya sesuai dengan kondisi yang sesuai, kita dapat menghitung nilai L^* , a^* , dan b^* dalam ruang warna Lab^* dari nilai XYZ dan White Reference-nya. Nilai L^* , a^* , dan b^* akan memberikan representasi tiga dimensi dari ruang warna, yang memungkinkan representasi warna yang lebih sesuai dengan persepsi manusia dan sering digunakan dalam pengolahan citra dan analisis warna..

$$f(w) = w^{\frac{1}{3}} \quad \text{for } w > 0.008856$$

$$f(w) = 7.787(w) + 0.1379 \quad \text{for } 0.0 \leq w \leq 0.008856 \quad (3)$$

Pada tahap akhir perhitungan untuk mengubah ruang warna XYZ menjadi Lab^* , terdapat dua kondisi yang harus diperhatikan sesuai dengan persamaan(4). Pertama, ketika nilai Y/Y_0 lebih besar dari 0.008856. Kedua, ketika nilai Y/Y_0 berada pada rentang 0 hingga 0.008856. Berikut contoh perhitungan konversi ruang warna Lab^* untuk piksel (i,j) :

$$L^* = 116 \left(\frac{Y}{Y_0} \right)^{\frac{1}{3}} - 16 \quad \text{for } \frac{Y}{Y_0} > 0,008856$$

$$L^* = 903,3 \frac{Y}{Y_0} \quad \text{for } 0,0 \leq \frac{Y}{Y_0} \leq 0,008856$$

$$a^* = 500 \left(f\left(\frac{X}{X_0}\right) - f\left(\frac{Y}{Y_0}\right) \right) \quad (4)$$

$$b^* = 200 \left(f\left(\frac{Y}{Y_0}\right) - f\left(\frac{Z}{Z_0}\right) \right)$$

Pada tahap akhir dari proses *pre-processing*, citra asli yang telah di-*rescaling* dan diubah ke ruang warna Lab^* akan disimpan dalam bentuk tabel matrix. Kemudian, citra tersebut akan digunakan

sebagai *input* untuk proses seleksi fitur dan proses pembelajaran dengan menggunakan algoritma K-Means secara manual.

Manualisasi Segmentasi Citra Menggunakan K-Means

untuk melakukan segmentasi citra menggunakan algoritma K-Means, pertama-tama perlu dilakukan pemilihan nilai *centroid* awal. Pemilihan *centroid* awal ini bertujuan untuk menentukan garis pemisah pada proses seleksi data menggunakan garis *ellipse* yang mengelilingi setiap *centroid*. Pada tahap ini, *centroid* awal yang dipilih menggunakan nilai a^* dan b^* terbanyak dari setiap piksel. Kemudian, dilakukan proses penentuan garis *ellipse* dengan menghitung jarak terdekat dari setiap piksel ke *centroid* yang dipilih. Setelah itu, dilakukan proses pemilihan *centroid* baru dengan mengambil rata-rata a^* dan b^* dari setiap piksel yang tergabung dalam garis *ellipse* tersebut. Proses ini terus dilakukan hingga tidak ada lagi perubahan *centroid*.

Manualisasi Segmentasi Daun Bawang

Segmentasi daun dilakukan dengan menggunakan algoritma K-Means dengan jumlah *cluster* yang telah ditentukan sebanyak 2. Data diolah dengan menggabungkan setiap bagiannya dan memberikan *index* pada setiap piksel. Untuk memulai proses segmentasi daun menggunakan algoritma K-Means, langkah pertama yang dilakukan adalah menentukan nilai *centroid* awal sebanyak jumlah *cluster* yang telah ditentukan sebanyak 2. Nilai *centroid* awal tersebut ditentukan secara acak,. Algoritma K-Means akan menggunakan nilai *centroid* tersebut sebagai dasar untuk mengelompokkan data pada setiap iterasinya.

Centroid awal yang terpilih secara acak adalah **piksel** (i,j) dan **piksel** (i,j) . Kemudian proses selanjutnya adalah menentukan titik-titik yang tergabung dengan setiap *centroid*. Titik-titik yang tergabung ditentukan dengan menghitung jarak antara titik tersebut dengan setiap *centroid*. Jarak antara titik tersebut dengan *centroid* dapat dihitung

menggunakan rumus *Euclidean Distance* dengan persamaan(5). Berikut contoh perhitungan jarak antara titik (i,j) dengan setiap *Centroid*:

$$d_{c1} = \sqrt{(c_1a^* - c_{c1}a^*)^2 + (c_1b^* - c_{c1}b^*)^2}$$

$$d_{c2} = \sqrt{(c_1a^* - c_{c2}a^*)^2 + (c_1b^* - c_{c2}b^*)^2} \quad (5)$$

Pada iterasi ke 0, hasil perhitungan jarak antara setiap titik dengan *centroid* c1 dan c2. Berdasarkan hasil perhitungan jarak antara titik (0,0) dengan kedua *centroid* tersebut, maka titik (0,0) diputuskan untuk masuk kategori c1. Hasil perhitungan jarak untuk titik lainnya juga dilakukan secara serupa untuk menentukan kategori *centroid* yang sesuai bagi setiap titik.

Pada tahap selanjutnya dari proses segmentasi daun menggunakan algoritma *K-Means*, dilakukan pembaharuan *centroid* dengan menghitung rata-rata dari tiap variabel a* dan b* pada setiap *cluster* yang sama. Berdasarkan, terdapat jumlah *cluster* 1 sebanyak x dan jumlah *cluster* 2 sebanyak x. Nilai *centroid* baru dapat dihitung dengan menggunakan rumus rata-rata yaitu menjumlahkan nilai a* dan b* pada setiap *cluster* yang sama kemudian dibagi dengan jumlah total piksel pada *cluster* tersebut. Berikut contoh perhitungan *centroid* baru pada iterasi ke 0:

Untuk mengetahui nilai ac1 dan bc1, Untuk mengetahui nilai ac2 dan bc2, digunakan rumus persamaan (6):

$$a_{c1} = \left(\frac{(\sum a \text{ pada cluster 1})}{\text{jumlah data pada cluster 1}} \right)$$

$$b_{c1} = \left(\frac{(\sum b \text{ pada cluster 1})}{\text{jumlah data pada cluster 1}} \right) \quad (6)$$

$$a_{c2} = \left(\frac{(\sum a \text{ pada cluster 2})}{\text{jumlah data pada cluster 2}} \right)$$

$$b_{c2} = \left(\frac{(\sum b \text{ pada cluster } 2)}{\text{jumlah data pada cluster } 2} \right)$$

Berdasarkan hasil perhitungan tersebut, didapatkan nilai *centroid* baru untuk iterasi ke 1, *centroid* yang digunakan untuk perhitungan jarak antara setiap data dengan *centroid* dilakukan sesuai dengan yang telah dijelaskan pada iterasi sebelumnya.

Pada tahap ini jika, algoritma K-Means telah mencapai konvergensi. Artinya, tidak ada kelas *cluster* yang berpindah pada iterasi ke 1. Hasil akhir dari Algoritma *K-Means* yang menyatakan jumlah data pada setiap kelas *cluster*.

6. Otsu

Metode *Otsu* digunakan untuk secara otomatis membagi histogram citra keabuan menjadi dua daerah yang berbeda tanpa memerlukan bantuan pengguna untuk memasukkan nilai ambang secara manual. Pendekatan yang digunakan oleh metode *Otsu* adalah analisis diskriminan, yang menentukan suatu variabel untuk membedakan antara dua atau lebih kelompok yang muncul secara alami. Dalam konteks memisahkan objek dengan latar belakang, analisis diskriminan akan memaksimalkan variabel tersebut. Untuk mendapatkan nilai ambang (*threshold*), terdapat perhitungan yang harus dilakukan. Langkah awal adalah membuat histogram, yang memberikan informasi tentang jumlah piksel untuk setiap tingkat keabuan. (Putra, 2004).

Pemisahan objek dan latar belakang serta pengukuran tingkat keabuan melalui sistem tertentu adalah apa yang disebut sebagai

histogram. Perhitungan mengenai distribusi tingkat keabuan suatu citra dapat dilakukan menggunakan rumus berikut ini:

$$P_i = \frac{n_i}{N} \dots\dots\dots (7)$$

Keterangan :

Pi = Probabilitas piksel ke-i

Ni = Jumlah piksel berintensitas i

N = Jumlah piksel pada citra

7. Python

Python adalah bahasa pemrograman tingkat tinggi yang populer, yang dapat menangani berbagai tugas pemrograman seperti perhitungan numerik, pengembangan web, pemrograman *database*, pemrograman jaringan, pemrosesan paralel, dan lainnya. Python menjadi populer karena beberapa alasan termasuk gratis, tersedia di semua sistem operasi populer, mudah dibaca dan di-*debug*, memiliki banyak modul yang bisa di-*install*, mudah di-*impor* ke platform lain, dan memiliki komunitas pengembang dan pengguna yang terdedikasi. Ada beberapa lingkungan python yang tersedia, termasuk Enthought Python Distribution dan PythonXY(Chityala dan Pudipeddi, 2014).

8. OpenCV

OpenCV adalah library yang digunakan untuk pemrosesan citra dan pengolahan sinyal video. Library ini menyediakan berbagai macam fungsi dan algoritma yang dapat digunakan untuk pemrosesan citra, seperti pembersihan citra, ekstraksi fitur, dan segmentasi citra. OpenCV memiliki banyak kelebihan dibandingkan dengan library lainnya, seperti : dukungan yang luas untuk sistem operasi, mudah digunakan dan didokumentasikan, serta tersedia dalam banyak bahasa pemrograman.(Chityala dan Pudipeddi, 2014)

9. Framework Django

Django adalah *framework web* yang dibangun menggunakan bahasa pemrograman python dan digunakan untuk membangun aplikasi web yang berkualitas tinggi. Django membantu kita untuk menghindari tugas-tugas yang terus berulang saat membangun sebuah aplikasi web, sehingga memudahkan proses pengembangan web dan menghemat waktu(Curtin, 2016). Filosofi desain Django antara lain:

- a) Longgar Terikat (*Loosely Coupled*) - Dalam Django, tujuan adalah untuk membuat setiap elemen stack se-independen mungkin.
- b) Pembangunan Cepat (*Fast Development*) - Tujuan adalah melakukan apapun yang diperlukan untuk memfasilitasi pembangunan aplikasi web yang cepat.
- c) Kode Lebih Sedikit (*Less Coding*) - Kode yang lebih sedikit digunakan untuk memfasilitasi pembangunan yang cepat.
- d) Desain Bersih (*Clean Design*) - Tujuan Django adalah untuk mengeluarkan desain yang sangat bersih untuk memudahkan pengembangan.
- e) Strategi-strategi yang jelas digunakan selama proses pengembangan web.
- f) Jangan Ulangi Diri Sendiri (*Don't Repeat Yourself (DRY)*) - Semuanya di Django harus dikembangkan di satu tempat, bukan mengulanginya berulang-ulang.

10. Tailwindcss

Tailwind CSS adalah sebuah framework CSS yang dirancang untuk mempermudah pembuatan antar muka pengguna (UI) yang responsif dan cepat. Framework ini menggunakan pendekatan utility-first, yang berarti setiap kelas CSS yang disediakan oleh Tailwind digunakan secara langsung dalam markup HTML, tanpa perlu menulis CSS khusus(Tailwind Labs Inc, 2023).Pada penelitian ini, Tailwind CSS dapat digunakan untuk membangun antar muka pengguna yang responsif dan mempercepat pengembangan aplikasi identifikasi daun bawang merah.

Dengan memanfaatkan utility classes dan kemampuan responsifnya, Tailwind CSS dapat membantu dalam mengatur tampilan elemen-elemen UI secara efisien dan sesuai dengan kebutuhan proyek.

11. *MSE dan PSNR*

MSE adalah nilai error dan semakin kecil nilainya, semakin mirip kedua gambar tersebut. Namun, PSNR adalah metrik yang mengukur kualitas gambar dengan mempertimbangkan MSE dan level intensitas maksimum (L) gambar. PSNR akan memberikan informasi tentang seberapa baik citra hasil pemrosesan dibandingkan dengan citra asli dalam hal rasio signal-to-noise. Semakin tinggi nilai PSNR, semakin baik kualitas gambar hasil pemrosesan.

Berikut adalah rumus matematis:

$$MSE = (1/(N*M)) * \sum[\sum(I(i,j) - K(i,j))^2]$$

$$PSNR = 10 * \log_{10}((L^2) / MSE)$$

Dengan menggunakan langkah-langkah di atas, kita dapat menggabungkan MSE dan PSNR menjadi satu metrik yang menggambarkan kualitas atau kesamaan antara dua gambar. Semakin tinggi nilai PSNR, semakin mirip atau berkualitas tinggi gambar hasil pemrosesan dibandingkan dengan gambar asli.

B. Kajian Pustaka

Dalam penelitian ini, akan dilakukan tinjauan studi yang mendukung penelitian tersebut. Beberapa tinjauan studi yang akan digunakan adalah sebagai berikut:

1. Penelitian ini tentang Sistem Pakar Diagnosis Hama dan Penyakit Bawang Merah Menggunakan Metode Dempster Shafer oleh Aldo (2020), Hasil penelitian menunjukkan bahwa menggunakan metode Dempster Shafer, terdapat 10 data serangan yang diolah. Metode ini dapat menghasilkan diagnosis jenis hama dan penyakit pada bawang

merah beserta langkah-langkah penanganannya, dengan tingkat akurasi mencapai 95%. Oleh karena itu, metode ini cocok digunakan dalam diagnosis hama dan penyakit pada tanaman bawang merah.

2. Penelitian ini tentang Segmentasi Citra Penyakit Pada Batang Buah Naga Menggunakan Metode Ruang warna L^*A^*B oleh Hakim dkk., (2020), Hasil penelitian menunjukkan bahwa menggunakan metode yang diusulkan, citra hasil segmentasi objek citra berpenyakit memperoleh tingkat akurasi sebesar 92.63%. Hal ini menunjukkan bahwa algoritma yang diusulkan mampu melakukan segmentasi objek citra berpenyakit dengan baik.
3. Penelitian ini tentang Analisis Perbandingan Segmentasi Citra Menggunakan Metode K-Means dan Fuzzy C-Means oleh Andika & Anisa (2020), Hasil penelitian menunjukkan bahwa metode K-Means merupakan metode yang sederhana dan memiliki perhitungan yang cepat. Sebelum melakukan segmentasi atau identifikasi pada daun, dilakukan penentuan ruang warna menggunakan CIELab. Selain itu, identifikasi data uji dilakukan dengan dua pendekatan, yaitu analisis bentuk dan analisis tekstur.
4. Penelitian ini tentang Segmentasi Citra Paru Menggunakan Metode k Means Clustering oleh Andika & Anisa (2020), menunjukkan bahwa K-Means merupakan metode yang simple dan cepat perhitungannya, Sebelum melakukan segmentasi atau identifikasi pada daun terlebih dahulu menentukan ruang warna menggunakan CIELab. Identifikasi data test menggunakan dua pendekatan, yaitu analisis bentuk dan analisis tekstur.
5. Penelitian ini tentang pengenalan citra bunga menggunakan segmentasi otsu threshold dsn naive bayes oleh Perani Rosyani, Oke Hariansyah (2020), menunjukkan bahwa Tujuan penelitian ini adalah meningkatkan tingkat akurasi dalam proses pengenalan objek dibandingkan dengan penelitian sebelumnya. Hal ini dicapai melalui proses segmentasi citra menggunakan metode Otsu Thresholding dan

penerapan operasi citra lainnya untuk menghilangkan noise, sehingga objek yang dihasilkan menjadi lebih bersih tanpa adanya gangguan noise. Hasil dari klasifikasi menggunakan metode Naïve Bayes menunjukkan tingkat akurasi yang lebih tinggi, yaitu sebesar 99.168%, dengan relative absolute error sebesar 8.0937%. Hasil ini secara signifikan lebih baik daripada penelitian sebelumnya yang mencapai tingkat akurasi sebesar 83.83%.

6. Penelitian tentang Segmentasi Berbasis K-Means Pada Deteksi Citra Penyakit Daun Tanaman Jagung dilakukan oleh Ulla Delfana Rosiani dkk., (2020) Penggunaan metode pengolahan citra digital dan segmentasi K-Means telah terbukti efektif dalam mendeteksi penyakit daun jagung, dengan tingkat akurasi sebesar 98,33%.