

KLASIFIKASI TUMBUHAN *ANGIOSPERMAE* MENGGUNAKAN ALGORITMA *K-NEAREST NEIGHBOR* BERDASARKAN PADA BENTUK DAUN

Arvin Christopher¹⁾, Teady Matius Surya Mulyana²⁾

^{1,2)}Program Studi Informatika, Universitas Bunda Mulia

Jl. Lodan Raya No. 2, Ancol, Jakarta Utara

e-mail: arvin.christopher27@gmail.com¹⁾, tmulyana@bundamulia.ac.id²⁾

ABSTRAK

Sekitar 250.000 spesies tanaman yang hidup merupakan tanaman kelompok Angiospermae. Keberagaman tanaman Angiospermae di Indonesia menimbulkan kesulitan bagi seseorang dalam melakukan proses identifikasi tanaman masuk ke kelas monokotil atau dikotil. Hal tersebut yang melatarbelakangi dilakukannya penelitian ini dengan mengklasifikasikan tanaman Angiospermae dengan menggunakan algoritma *K-Nearest Neighbor*. Penelitian terkait pengklasifikasian tanaman manga menggunakan algoritma *Backpropagation* dengan pengekstraksian fitur menggunakan *GLCM* menghasilkan akurasi sebesar 49%. Selanjutnya, pengklasifikasian tanaman menggunakan algoritma *K-Nearest Neighbor* dengan pengekstraksian fitur morfologi daun menghasilkan akurasi sebesar 92%. Terdapat juga pengklasifikasian tanaman jeruk menggunakan algoritma *K-Nearest Neighbor* dengan pengekstraksian fitur tekstur daun menghasilkan akurasi sebesar 81,48%. Pada penelitian ini, digunakan algoritma pengklasifikasian *K-Nearest Neighbor* dengan pengekstraksian fitur menggunakan matriks populasi dengan tujuan untuk meneliti tingkat akurasi pengklasifikasian menggunakan pengekstraksian fitur matriks populasi. Data yang digunakan adalah citra digital daun yang akan diekstrak fitur bentuknya menggunakan fitur matriks populasi dengan melakukan *image pre-processing* dan pendeteksian terlebih dahulu. Data tersebut kemudian dibagi menjadi data training sebesar 70%, dan data test sebesar 30%. Hasil dari penelitian ini adalah nilai akurasi pengujian data test yang didapatkan sebesar 81% dengan jumlah tetangga $k=1$. Setelah dilakukan pengujian, akan dilakukan evaluasi model dengan menggunakan *confusion matrix*, yang terdiri dari *accuracy*, *precision*, *recall*, dan *f-1 score*.

Kata Kunci: *angiospermae*, *canny*, *confusion matrix*, *k-nearest neighbor*, *matriks populasi*

ABSTRACT

About 250,000 living plant species are plants of the Angiosperms group. The diversity of Angiosperm plants in Indonesia makes it difficult for someone to carry out the process of identifying plants into the monocotyledonous or dicotyledonous class. This is the background for doing this research by classifying Angiosperms plants using the *K-Nearest Neighbor* algorithm. Research related to classifying manga plants using the *Backpropagation* algorithm with feature extraction using *GLCM* produces an accuracy of 49%. Furthermore, the classification of plants using the *K-Nearest Neighbor* algorithm with leaf morphological features extraction resulted in an accuracy of 92%. There is also a classification of citrus plants using the *K-Nearest Neighbor* algorithm with leaf texture feature extraction resulting in an accuracy of 81.48%. In this study, the *K-Nearest Neighbor* classification algorithm is used with shape feature extraction using a population matrix with the aim of examining the level of classification accuracy using population matrix feature extraction. The data used is a digital leaf image which will extract its shape features using a population matrix feature by doing *image pre-processing* and detection first. The data is then divided into training data of 70%, and test data of 30%. The results of this study are the accuracy of the test data obtained by 81% with the number of neighbors $k=1$. After testing, an evaluation of the model will be carried out using a *confusion matrix*, which consists of *accuracy*, *precision*, *recall*, and *f-1 score*.

Keywords: *angiospermae*, *canny*, *confusion matrix*, *k-nearest neighbor*, *population matrix*

I. PENDAHULUAN

INDONESIA merupakan salah satu negara yang menjadi salah satu pusat keanekaragaman hayati, atau yang biasa disebut sebagai negara megabiodiversity [1]. Berdasarkan data dari Menteri Lingkungan Hidup dan Kehutanan, luas kawasan hutan di Indonesia pada Desember 2021 mencapai 95,6 juta hectare, dengan 46,9 juta hectare hutan primer, 43,1 juta hectare hutan sekunder, dan 5,4 juta hectare hutan tanaman [2]. Dari sekian banyaknya tanaman, terdapat 2 jenis tanaman, yaitu tanaman tidak berbiji dan berbiji. Tanaman berbiji (*Spermatophyta*) dibagi lagi menjadi 2, yaitu tanaman Angiospermae dan Gymnospermae. Angiospermae berasal dari 2 buah kata dalam bahasa Yunani, yaitu “angion” yang berarti tertutup dan “spermae” yang berarti biji, sehingga angiosperm, *canny*, *confusion matrix*, *k-nearest neighbor*, *population matrix*. Angiospermae memiliki arti tanaman berbiji tertutup [3]. Saat ini, terdapat sekitar 250.000 spesies tanaman dari kelompok tanaman Angiospermae. Angiospermae terdiri dari 2 kelas, yaitu kelas monokotil dan dikotil [4]. Kedua kelas tersebut dapat

dibedakan dari masing-masing ciri kelasnya, salah satunya adalah bentuk tepi dan pertulangan daun.

Namun, dengan berbagai keberagaman tanaman tersebut menyebabkan kesulitan dalam proses identifikasi untuk membedakan setiap tumbuhan termasuk ke dalam kelas monokotil atau ke dalam kelas dikotil. Oleh karena itu, Machine Learning menjadi salah satu solusi yang diharapkan dapat membantu mempermudah manusia dalam mengidentifikasi sebuah tanaman termasuk ke dalam kelas monokotil atau dikotil yang didasarkan dari bentuk pertulangan daun dan tepi daunnya.

Beberapa penelitian terkait pengklasifikasian berdasarkan citra daun telah dilakukan sebelumnya, beberapa penelitian tersebut adalah “Implementasi Algoritma Canny dan Backpropagation Untuk Mengklasifikasi Jenis Tanaman Mangga” oleh Elvia Budianita, Tricia Ulfadhyan, Febi Yanto, dan Pizaini [5] mendapatkan hasil akurasi sebesar 49% dengan menggunakan metode pengklasifikasian *Backpropagation* disertai dengan pengekstraksian fitur GLCM, bertujuan untuk mengklasifikasikan tanaman mangga sesuai dengan jenisnya berdasarkan pada bentuk daun dikarenakan masyarakat mengetahui jenis tanaman mangga berdasarkan pada buahnya saja, yang dimana buah bergantung pada musim. Selanjutnya, “Sistem Klasifikasi Jenis Tanaman Hias Daun Philodendron Menggunakan Metode K-Nearest Neighbor (KNN) Berdasarkan Nilai Hue, Saturation, Value (HSV)” oleh Dani Syahid, Jumadi, dan Dian Nursantika [6] dengan tujuan untuk mengetahui nama tumbuhan yang diminati dari sekian banyaknya tanaman hias, mendapatkan hasil akurasi sebesar 92% dengan menggunakan metode pengklasifikasian K-Nearest Neighbor disertai dengan pengekstraksian fitur warna HSV. Lalu, “Klasifikasi Citra Daun dengan GLCM (Gray Level Co-Occurrence) dan K-NN (K-Nearest Neighbor)” oleh Sri Ayu Rosiva Srg, Muhammad Zarlis, dan Wanayumini [7] mendapatkan hasil akurasi sebesar 98% menggunakan metode pengklasifikasian K-Nearest Neighbor disertai dengan pengekstraksian fitur tekstur GLCM, memiliki tujuan untuk membangun sistem klasifikasi tanaman berdasarkan citra daun yang akurat dan tingkat kesalahan yang minimal.

Dibalik sebuah penelitian, terdapat sebuah kelemahan ataupun keterbatasan. Penelitian [5] menggunakan pendeteksian tepi *canny* yang dapat berfungsi dengan baik, namun mendapatkan akurasi pengklasifikasian yang rendah menggunakan metode klasifikasi *Backpropagation* dengan menggunakan data scanner sebanyak 450 data dengan pembagian data latih : data uji sebesar 90%:10%, disertai dengan penggunaan parameter *learning rate* 0,1;0,01; dan 0,001. Selanjutnya adalah penelitian [6] yang memiliki 4 hasil pengklasifikasian yang tidak sesuai dengan kategori dikarenakan intensitas cahaya citra yang berbeda dengan data latih. Kemudian, penelitian [7] mendapatkan 2 hasil pengklasifikasian yang salah dikategorikan yang disebabkan beberapa hasil fitur GLCM yang mirip dan proses akuisisi citra yang belum sempurna.

Novelty dari penelitian ini adalah pengklasifikasian dengan menggunakan algoritma KNN, sedangkan penelitian [5] melakukan pengklasifikasian menggunakan metode *Backpropagation*. *Novelty* kedua dari penelitian ini adalah pengekstraksian fitur bentuk menggunakan fitur matriks populasi, sedangkan penelitian [6] menggunakan pengekstraksian fitur warna HSV (*Hue, Saturation, Value*) dan penelitian [7] menggunakan pengekstraksian fitur tekstur, yaitu GLCM.

Pada penelitian ini, digunakan sebuah algoritma Machine Learning, yaitu *K-Nearest Neighbors* atau yang biasa disebut KNN. KNN merupakan sebuah algoritma pengklasifikasian antara suatu data berdasarkan objek yang terdekat. KNN juga merupakan algoritma pengklasifikasian dengan rumus yang cukup sederhana sehingga mudah untuk diaplikasikan. Penggunaan algoritma KNN dalam penelitian ini didasari oleh hasil akurasi yang cukup tinggi, mampu mengolah data training yang memiliki banyak noise, dan efektif jika jumlah data training yang digunakan besar [8]. Agar algoritma KNN dapat dijalankan dengan baik, maka diperlukan sebuah teknik pengolahan citra yang digunakan untuk menunjukkan sebuah fitur atau atribut dari sebuah objek pada citra, yaitu deteksi tepi. Deteksi tepi merupakan salah satu teknik pengolahan citra yang digunakan untuk mempresentasikan objek-objek yang terkandung pada sebuah citra [9] dengan mengidentifikasi garis batas suatu objek dengan latar belakang [10]. Terdapat beberapa metode deteksi tepi, yaitu metode *Sobel, Prewitt, Robert* dan *Canny* [11]. Dalam penelitian ini, digunakan metode deteksi tepi *Canny*. Teknik deteksi tepi merupakan langkah awal dalam melakukan ekstraksi fitur.

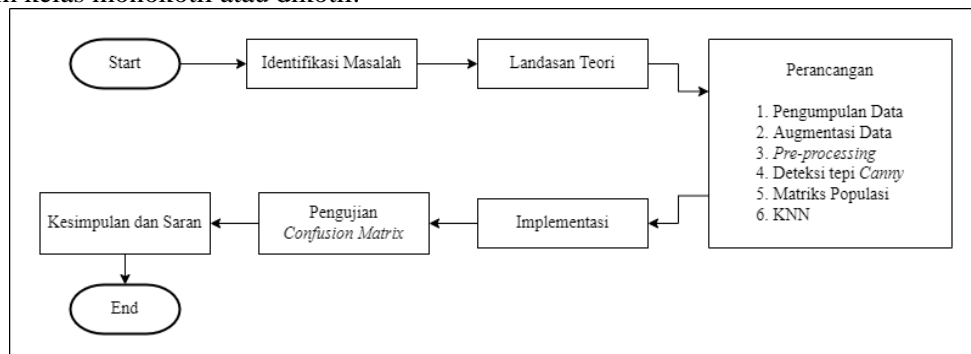
Ekstraksi fitur dilakukan dengan menggunakan fitur matriks populasi. Matriks populasi merupakan salah satu teknik pengekstraksian fitur yang dilakukan dengan cara menghitung populasi piksel dari objek pada sebuah citra. Fitur yang diekstrak adalah fitur bentuk daun yang direpresentasikan dengan bentuk tepi dan pertulangan daun. Fitur tersebut akan diproses untuk dilakukan pengenalan terhadap objek lainnya menggunakan metode klasifikasi KNN. Proses pengklasifikasian dilakukan dengan menghitung jarak terdekat antara fitur data *testing* dengan fitur data *training* lalu dilakukan *voting* dengan jumlah tetangga (k) yang telah ditentukan.

II. METODE PENELITIAN

Tahapan atau metodologi yang digunakan dalam penelitian ini dapat dilihat pada Gambar 1. Tahapan penelitian ini dimulai dari identifikasi masalah, mencari landasan teori, melakukan perancangan, implementasi pada program dan data, Pengujian dengan Confusion Matrix dan mengambil kesimpulan.

A. Identifikasi Masalah

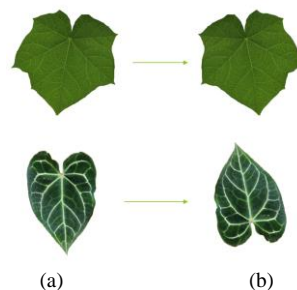
Peneliti mengidentifikasi sebuah masalah, yaitu terjadinya kesulitan dalam mengidentifikasi sebuah tanaman masuk ke dalam kelas monokotil atau dikotil.



Gambar 1. Tahapan penelitian

B. Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah citra digital daun berjumlah 2000 citra yang terdiri dari 2 kelas, yaitu 1000 citra digital daun kelas monokotil, dan 1000 citra digital daun kelas dikotil. Citra diambil dengan menggunakan kamera *handphone* Xiaomi Poco F3 dengan kamera belakang 48 MP. Ukuran piksel citra digital daun yang digunakan adalah 3000x3000 piksel.



Gambar 2. (a) Citra daun sebelum dilakukan augmentasi, (b) Citra daun setelah dilakukan augmentasi

C. Pemilihan Metode

Penggunaan algoritma KNN dalam penelitian ini didasari oleh pengklasifikasian tanaman Angiospermae yang memiliki 2 kelas tanaman, yaitu kelas tanaman monokotil dan dikotil yang dapat diklasifikasikan berdasarkan dari fitur bentuk, yaitu tepi daun dan pertulangan daun. Selain itu, penggunaan algoritma KNN dalam penelitian ini juga digunakan untuk meneliti nilai akurasi pengklasifikasian dengan pengekstraksian fitur bentuk daun menggunakan fitur matriks populasi.

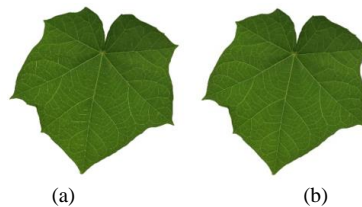
D. Augmentasi Data

Augmentasi data dilakukan dengan memodifikasi data gambar atau citra yang sudah ada sehingga komputer akan mendeteksi citra tersebut sebagai citra baru atau berbeda dari citra sebelumnya [12]. Pada penelitian ini, data diaugmentasi dengan melakukan rotasi dan juga *flip* atau membalikkan gambar secara horizontal dan vertikal. Contoh citra sebelum dan setelah dilakukan augmentasi dapat dilihat pada Gambar 2 dimana citra daun sebelum dilakukan augmentasi ditunjukkan pada Gambar 2 (a) dan citra daun setelah dilakukan augmentasi ditunjukkan pada Gambar 2 (b).

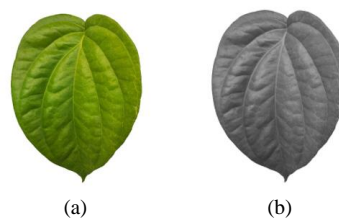
E. Pre-processing

Pada penelitian ini, *pre-processing* terdiri dari 2 tahap, yaitu *resize* dan *grayscale*. Citra yang akan digunakan dalam penelitian ini memiliki ukuran yang cukup besar. Ukuran citra yang besar membuat sistem memiliki kinerja yang lebih lambat. Untuk mempercepat kinerja sistem, dibuat standar ukuran yang tepat dan citra yang memiliki ukuran yang tidak sesuai akan diubah menjadi seukuran standar sehingga dapat mengurangi detail yang berlebih dari citra yang digunakan dan dengan demikian akan memberikan hasil yang lebih akurat. Ukuran standar pada penelitian ini ialah 256x256 pixel. Citra sebelum dan setelah dilakukan tahap *resize* dapat dilihat pada Gambar 3 dimana citra sebelum tahap *resize* dengan ukuran 3000x3000 piksel ditunjukkan pada Gambar 3 (a) dan citra setelah tahap *resize* dengan ukuran 256x256 piksel ditunjukkan pada Gambar 3 (b).

Citra berwarna yang akan digunakan dalam penelitian diubah terlebih dahulu menjadi citra abu-abu. Untuk mengubah citra RGB menjadi *grayscale*, dapat dilakukan dengan mengambil informasi 3 warna dasar, yaitu *red*, *green*, dan *blue* dari setiap piksel citra. 3 warna dasar tersebut akan dikalikan dengan masing-masing bobot nilai komposisi setiap warna, seperti yang ditunjukkan pada persamaan (2). Nilai perkalian tersebut kemudian akan dijumlahkan, yang nantinya hasil penjumlahan tersebut menunjukkan tingkat keabuan pada setiap piksel citra sesuai dengan nilai yang didapatkan, sehingga citra RGB dapat berubah menjadi citra *grayscale*.



Gambar 3. (a) Citra daun sebelum tahap *resize*, (b) Citra daun setelah tahap *resize*



Gambar 4. (a) Citra daun asli sebelum tahap *grayscale*, (b) Citra daun setelah tahap *grayscale*

Konversi citra warna menjadi citra *grayscale* juga dapat dilakukan menggunakan *weighted method*. *Weighted method* memiliki persamaan yang dapat dilihat pada persamaan (1).

$$W_r + W_g + W_b = 1 \quad (1)$$

Dimana:

W = bobot nilai komposisi setiap warna pada piksel tersebut (*Red*, *Green*, *Blue*).

Nilai W_r , W_g , dan juga W_b merupakan koefisien dengan setiap warnanya berdasarkan urutan yaitu 0.299, 0.587, dan 0.114 sehingga didapatkan persamaan (2) [13].

$$Gray = 0.299 * Red + 0.587 * Green + 0.114 * Blue \quad (2)$$

Citra daun sebelum dan sesudah dilakukan tahap *grayscale* dapat dilihat pada Gambar 4 dimana citra daun asli sebelum dilakukan tahap *Grayscale* ditunjukkan pada Gambar 4 (a) dan citra daun setelah dilakukan tahap *Grayscale* pada Gambar 4 (b).

F. Deteksi Tepi Canny

Deteksi tepi merupakan sebuah proses pengolahan citra yang digunakan untuk mendeteksi sebuah objek yang

terdapat pada sebuah citra dengan memberikan garis tepi pada sebuah objek berdasarkan dari nilai keabuan sebuah objek dengan latar belakang [14]. Metode deteksi tepi yang digunakan pada penelitian ini adalah deteksi tepi *canny*. Deteksi tepi *canny* adalah deteksi tepi yang digunakan untuk mendeteksi tepi objek dengan cara menghapus *noise* pada suatu citra dan menerapkannya pada fungsi *Gaussian Filter* dengan *double thresholding* [15]. Metode ini berfungsi untuk mendeteksi garis tepi objek pada citra melalui berbagai tahapan, yaitu *Gaussian Filtering*, *Gradient Calculation*, *Non-Maximum Suppression*, *Double Threshold*, dan *Hysteresis Thresholding*.

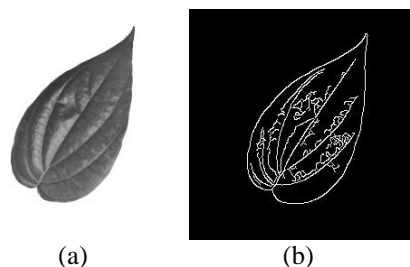
Gaussian Blur merupakan tahapan awal dari pendeteksian tepi *canny* yang digunakan untuk mengurangi *noise* atau derau pada citra daun. Untuk menerapkannya, tentukan ukuran dari kernel filter *Gaussian* terlebih dahulu. Pada sebuah citra, sering kali ditemukan *noise* dalam jumlah sedikit ataupun banyak. *Noise* tersebut membuat proses deteksi tepi menghasilkan *output* yang tidak tepat. Untuk menghindari hal tersebut, digunakanlah *Gaussian filter* yang akan terkonvolusi dengan gambar dan menghilangkan *noise*. Nilai *Gaussian* dapat dihitung menggunakan persamaan (3).

$$g(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3)$$

Menghitung gradien digunakan untuk mencari tepi objek dan arah tepi dengan cara menghitung gradien citra dan θ . Gradien dapat ditemukan dengan menggunakan operator deteksi tepi *Sobel*. Kemudian, gradien dan arah tepi dapat dihitung dengan menggunakan persamaan (4) dan (5).

$$G = \sqrt{G_x^2 + G_y^2} \quad (4)$$

$$\theta = \tan^{-1} \frac{G_y}{G_x} \quad (5)$$



Gambar 5. (a) Citra daun setelah tahap grayscale, (b) Citra daun setelah tahap deteksi tepi *canny*

Setelah mendapatkan arah tepi, maka langkah selanjutnya adalah proses *Non-Maximum Suppression*. *Non-Maximum Suppression* berfungsi untuk melakukan penipisan tepi objek. Nilai intensitas pada setiap pikselnya akan dibandingkan dengan 2 tetangga terdekatnya yang searah dengan arah tepi. Jika salah satu tetangganya memiliki nilai intensitas piksel yang lebih tinggi dari nilai intensitas piksel yang sedang diproses, maka piksel dengan nilai intensitas yang lebih tinggi akan dipertahankan, sedangkan piksel yang sedang diproses menjadi bernilai 0.

Proses yang terakhir adalah menggunakan 2 buah *threshold*, yaitu *low threshold* dan *high threshold*. *Low threshold* digunakan untuk mengidentifikasi piksel lemah sehingga nilai piksel tersebut menjadi 0 atau berwarna hitam. Sedangkan *high threshold* digunakan untuk mengidentifikasi piksel kuat sehingga nilai piksel tersebut menjadi 1 atau berwarna putih. Hasil pendeteksian tepi *canny* dapat dilihat pada Gambar 5 dimana citra daun setelah dilakukan tahap *grayscale* ditunjukkan pada Gambar 5 (a) dan citra daun setelah dilakukan tahap deteksi tepi *canny* ditunjukkan pada Gambar 5 (b).

G. Matriks Populasi

Kumpulan sel yang terbentuk dari pembagian citra yang membentuk kolom dan baris. Masing-masing sel tersebut berisikan populasi piksel dari objek pada sebuah citra. Matriks populasi piksel tersusun dari matriks dengan ukuran 2x2, 3x3, 4x4, dan seterusnya [16]. Matriks populasi piksel merupakan salah satu cara ekstraksi fitur, dimana fitur tersebut akan diproses untuk dilakukan pengenalan karakter [17].

Matriks populasi piksel digunakan untuk mengekstrak fitur bentuk tepi dan pertulangan daun. Hal yang pertama dilakukan adalah membentuk batas atas, kanan, bawah dan kiri objek dengan melakukan *scanning* baris per baris, dan kolom per kolom. Kemudian, citra tersebut dibagi berdasarkan ukuran matriks yang ditentukan. Jika ukuran matriks adalah 5x5, maka akan terdapat 25 sel. Setiap sel memiliki nilai persentase piksel objek yang ada pada sel tersebut yang dapat dihitung dengan mencari jumlah piksel yang mewakili objek citra dibagi dengan total jumlah piksel pada sel tersebut, dan dikali dengan 100%. Nilai-nilai persentase tersebut merupakan matriks populasi.

H. KNN

Metode *K-Nearest Neighbor* digunakan untuk mengklasifikasikan tanaman yang terdiri dari 2 kelas tanaman, yaitu monokotil dan dikotil. Metode *K-Nearest Neighbor* membutuhkan jumlah tetangga yang digunakan untuk mengklasifikasikan dengan cara menghitung jarak terdekat antara data *testing* dengan data *training* berdasarkan mayoritas jumlah tetangga yang telah ditentukan [18][19]. Perhitungan jarak tersebut menggunakan perhitungan *Euclidean Distance* dengan rumus pada persamaan (6).

$$d_{(i,j)} = \sqrt{(x_{i1} - x_{j1})^2 + \dots + (x_{in} - x_{jn})^2} \quad (6)$$

Notasi $d_{(i,j)}$ merupakan jarak data ke-i ke data ke-j, x_{i1} merupakan fitur dari data latih ke-1, dan x_{j1} merupakan fitur dari data uji ke-1 [20], n merupakan jumlah fitur yang digunakan. Hasil perhitungan jarak tersebut diurutkan dari yang terkecil hingga terbesar. Lalu dicari data yang memiliki hasil perhitungan jarak terdekat dengan jumlah tetangga yang telah ditentukan. Data *testing* dapat diklasifikasikan berdasarkan label dari data yang diambil sebanyak jumlah tetangga dengan jarak terdekat.

I. Confusion Matrix

Confusion Matrix adalah tabel yang menggambarkan performa dari sebuah algoritma atau model yang menyatakan jumlah data uji yang benar dan jumlah data uji yang salah dari proses pengklasifikasian [20]. Setiap kolom matriks merupakan kelas prediksi dari data, dan setiap baris matriks menyatakan kelas aktual dari data, ataupun sebaliknya. Matriks tersebut ditunjukkan pada Tabel 1.

TABEL 1
CONFUSION MATRIX

| | Predicted Positive | Predicted Negative |
|-----------------|---------------------|---------------------|
| Actual Positive | True Positive (TP) | False Negative (FN) |
| Actual Negative | False Positive (FP) | True Negative (TN) |

True Positive (TP) merupakan jumlah data dari kelas positif yang benar diklasifikasikan sebagai kelas positif. *True Negative* (TN) merupakan jumlah data dari kelas negative yang benar diklasifikasikan sebagai kelas negatif. *False Positive* (FP) merupakan jumlah data dari kelas negative yang salah diklasifikasikan sebagai kelas positif. *False Negative* (FN) merupakan jumlah data dari kelas positive yang salah diklasifikasikan sebagai kelas negatif.

Dari *confusion matrix* tersebut, terdapat 4 kategori data yang dapat digunakan untuk mengukur performa dari sebuah model atau algoritma, yaitu *Accuracy*, *Precision*, *Recall*, dan *F-1 Score*.

Accuracy merupakan persentase benar model dalam melakukan klasifikasi. Rumus *accuracy* ditunjukkan pada persamaan (7).

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \quad (7)$$

Precision merupakan persentase model memprediksi benar positif dengan keseluruhan data yang diprediksi positif. Rumus *precision* ditunjukkan pada persamaan (8).

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

Recall merupakan persentase prediksi benar positif ketika kelas aktual data tersebut positif. Rumus *recall* ditunjukkan pada persamaan (9).

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

F-1 Score merupakan rata-rata antara *precision* dengan *recall* yang dibobotkan. Rumus *f-1 score* ditunjukkan pada persamaan (10).

$$F1\ Score = \frac{(2*Recall*Precision)}{Recall+Precision} \quad (10)$$

III. HASIL DAN PEMBAHASAN

A. Persiapan Data

Data berupa citra digital daun sebanyak 2000 citra yang terdiri dari 2 kelas, yaitu 1000 citra digital daun kelas monokotil dan 1000 citra digital daun kelas dikotil. Data tersebut dibagi menjadi data *training* dan data *testing* sebesar 70% dan 30%, sehingga terdapat 1400 data *training* yang terdiri dari 700 citra daun monokotil dan dikotil, dan 600 data *testing* yang terdiri dari 300 citra daun monokotil dan dikotil. Latar belakang dari citra adalah warna putih. Contoh citra daun yang digunakan dalam penelitian ini dapat dilihat pada Gambar 6 yang menunjukkan contoh citra daun monokotil dan Gambar 7 yang menunjukkan contoh citra daun dikotil.



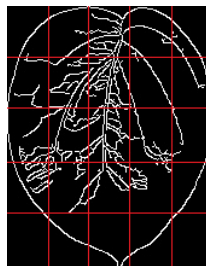
Gambar 6. Contoh citra digital daun monokotil



Gambar 7. Contoh citra digital daun dikotil

B. Hasil Ekstraksi Fitur

Data yang akan diekstrak fiturnya adalah data yang sudah melalui tahapan *pre-processing* (*resize* dan *grayscale*) dan pendeteksian tepi. Citra daun yang telah dilalui tahap pendeteksian tepi akan dilanjutkan ke tahap pengestraksian fitur menggunakan fitur matriks populasi. Contoh pembagian matriks 5x5 pada citra daun monokotil ditunjukkan pada Gambar 8 hasil pengestraksian fitur bentuk tepi dan pertulangan daun dari sebuah citra digital daun monokotil dapat dilihat pada Gambar 9.



Gambar 8 Pembagian matriks 5x5 pada citra daun monokotil

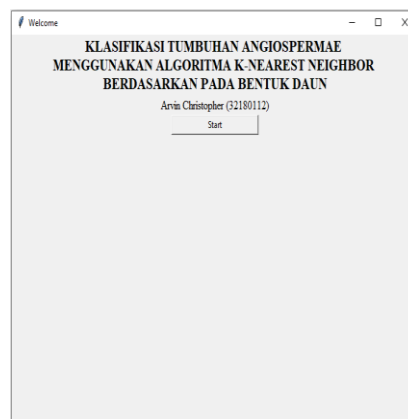
```
[ 3.94644116  5.9901339  19.73220578  8.80902044  7.89288231]
[ 8.31571529 15.6448203  24.59478506  5.77871741  5.07399577]
[10.64129669 16.34954193 21.84637068  8.52713178  4.01691332]
[13.88301621  9.09090909 10.78224101  7.04721635  5.00352361]
[ 0.35236082  3.87596899  3.45313601  4.08738548  1.33897111]
```

Gambar 9. Contoh hasil pengestraksian fitur bentuk tepi dan pertulangan daun

Hasil pengekstraksian fitur tersebut akan disimpan ke dalam sebuah file csv beserta dengan kelas dari masing-masing data, yang nantinya akan digunakan untuk melakukan *training* sehingga model dapat melakukan pengklasifikasian dengan cara menghitung jarak antara fitur data *testing* dengan fitur data *training*.

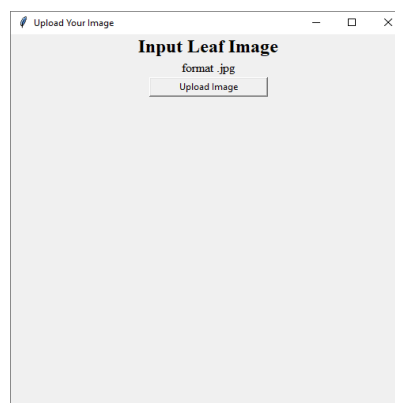
C. Klasifikasi K-Nearest Neighbor

File csv yang tersimpan akan digunakan untuk membagi data *training* sebesar 70% dan data *testing* sebesar 30% dengan menggunakan fungsi dari *library sklearn*. Setelah data berhasil dibagi, tahap selanjutnya adalah melakukan pengklasifikasian menggunakan fungsi *KNeighborsClassifier* dari *library sklearn* dengan menentukan terlebih dahulu jumlah tetangga yang akan digunakan. Jumlah tetangga yang digunakan dalam penelitian ini adalah $k=1$. Sebelum dilakukan pengujian, model di *training* terlebih dahulu menggunakan fungsi *fit* berdasarkan pada data *training* yang telah dibagi sebelumnya. Kemudian, dilakukanlah pengujian menggunakan fungsi *predict* untuk mengklasifikasikan tanaman *Angiospermae* dengan mengukur jarak fitur data *testing* dengan fitur data *training*. Tampilan halaman awal aplikasi dapat dilihat pada Gambar 10.



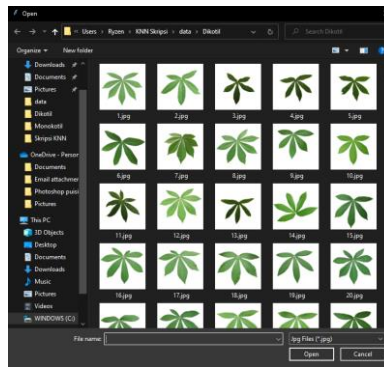
Gambar 10. Halaman Awal

Gambar 10 menunjukkan halaman awal dari aplikasi yang terdiri dari judul penelitian, nama dan NIM, dan juga tombol “*Start*” yang akan mengarahkan ke halaman *testing*. Tampilan halaman *testing* dapat dilihat pada Gambar 11.



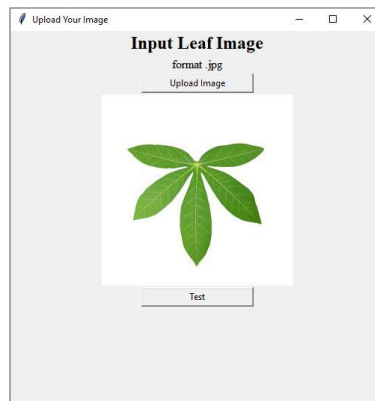
Gambar 11. Halaman *Testing* sebelum citra dimasukkan

Gambar 11 menunjukkan halaman *testing* dari aplikasi yang akan muncul setelah menekan tombol “*Start*” pada halaman awal. Halaman ini digunakan untuk melakukan *testing* dengan memasukkan citra daun terlebih dahulu dengan menekan tombol “*Upload Image*”. Dengan menekan tombol tersebut, akan muncul *desktop* untuk memilih citra yang akan digunakan, yang dapat dilihat pada Gambar 12.

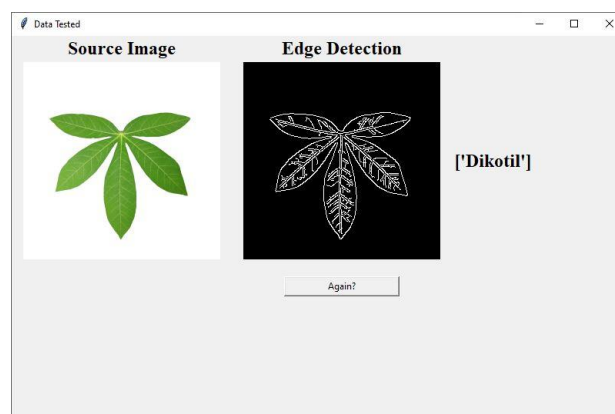


Gambar 12. Desktop memilih citra

Gambar 12 menunjukkan *desktop* pemilihan gambar. Setelah memilih citra yang akan digunakan, tekan tombol “Open” yang kemudian akan membawa pengguna ke halaman *testing* sebelumnya. Tampilan halaman *testing* setelah citra dimasukkan dapat dilihat pada Gambar 13.

Gambar 13. Halaman *Testing* setelah citra dimasukkan

Gambar 13 menunjukkan halaman *testing* setelah citra berhasil dimasukkan. Pada bagian bawah citra, terdapat tombol “Test” yang digunakan untuk mengklasifikasikan citra daun tersebut masuk ke dalam kelas monokotil atau dikotil. Tampilan halaman hasil *testing* dapat dilihat pada Gambar 14.

Gambar 14. Halaman Hasil *Testing*

Gambar 14 merupakan halaman setelah dilakukannya *testing* dengan menampilkan citra yang telah dimasukkan berada pada kotak “Source Image” dan hasil deteksi tepi dari citra yang telah dimasukkan berada pada kotak “Edge Detection”. Hasil prediksi citra daun tersebut dapat dilihat pada bagian kanan halaman hasil *testing*, yaitu Dikotil. Terdapat tombol “Again” yang digunakan untuk kembali ke halaman *testing* sebelum citra dimasukkan, sehingga pengguna dapat melakukan *testing* secara berulang sesuai dengan kebutuhannya.

D. Evaluasi Model

Untuk mengevaluasi model *K-Nearest Neighbor*, pada penelitian ini digunakan *confusion matrix* yang akan

menampilkan jumlah data yang benar dan salah diklasifikasi dengan jumlah tetangga $k=1$. *Confusion matrix* penelitian ini dapat dilihat pada Gambar 15. Berdasarkan *confusion matrix* pada Gambar 15, didapatkan laporan pengklasifikasian dari data *testing* yang berisikan nilai *precision*, *recall*, *f-1 score*, dan *accuracy*. Tabel hasil laporan klasifikasi dapat dilihat pada Tabel 2.

Berdasarkan pada Tabel 2, didapatkan nilai akurasi sebesar 81% dari pengklasifikasian menggunakan *K-Nearest Neighbor* dengan jumlah tetangga $k=1$. Berdasarkan nilai akurasi tersebut, dapat dinyatakan bahwa nilai akurasi dari penelitian ini lebih baik dari penelitian [5] dengan nilai akurasi sebesar 49%, tetapi lebih buruk dari penelitian [6] dan [7] dengan nilai akurasi berturut-turut sebesar 92% dan 98%.

Berdasarkan *confusion matrix* pada Gambar 15, kelas dikotil memiliki jumlah data yang benar diprediksi sebanyak 248 dan data yang salah diprediksi sebanyak 52 data. Sedangkan untuk kelas monokotil memiliki jumlah data yang benar diprediksi sebanyak 236 data dan data yang salah diprediksi sebanyak 64 data. Kesalahan dalam pengklasifikasian dapat terjadi karena beberapa faktor, seperti pemantulan cahaya pada badan daun, posisi pengakuisisi citra yang kurang baik dapat menyebabkan pertulangan daun tidak dapat dideteksi secara baik, dan juga adanya kemiripan antara daun yang ada pada kelas monokotil dan kelas dikotil sehingga pengeksktraksian menggunakan fitur matriks populasi memiliki nilai fitur yang mirip. Dengan kemiripan nilai fitur tersebut, maka metode *K-Nearest Neighbor* dapat mengklasifikasikan tanaman dikotil sebagai tanaman monokotil, dan begitupun juga sebaliknya.

| | | | |
|------------|-----------|-----------------|-----------|
| True Label | Dikotil | 248 | 52 |
| | Monokotil | 64 | 236 |
| | | Dikotil | Monokotil |
| | | Predicted Label | |

Gambar 15. Confusion Matrix

TABEL 2
CLASSIFICATION REPORT

| | Precision | Recall | F1-score |
|-----------|-----------|--------|----------|
| Dikotil | 0.79 | 0.83 | 0.81 |
| Monokotil | 0.82 | 0.79 | 0.80 |
| Accuracy | 0.81 | | |

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, didapatkan kesimpulan dari penelitian yang dimulai dari tahap *pre-processing*, deteksi tepi, ekstraksi fitur dan penggunaan metode *K-Nearest Neighbor* untuk mengklasifikasikan jenis tanaman *Angiospermae* dengan menggunakan citra daun, yaitu metode *K-Nearest Neighbor* mampu mengklasifikasikan tanaman *Angiospermae* (monokotil dan dikotil) dengan cukup baik. Penggunaan data *training* sebanyak 1400 data dan data *testing* sebanyak 600 data dengan menggunakan jumlah tetangga $k=1$, menghasilkan nilai akurasi sebesar 81%. Proses *pre-processing* dan deteksi tepi berfungsi dengan baik sehingga mampu memberikan hasil ekstraksi fitur yang baik. Terdapat beberapa faktor yang dapat mempengaruhi akurasi

pengklasifikasian, yaitu posisi saat mengakuisisi citra, pencahayaan yang kurang baik, dan bentuk daun yang mirip diantara kelas monokotil dan dikotil.

Untuk saran pengembangan penelitian ini, dapat dilakukan dengan menggunakan metode *preprocessing* dan deteksi tepi yang berbeda untuk membandingkan hasil pengekstraksian fitur menggunakan fitur matriks populasi. Selain itu, dapat menggunakan metode ekstraksi fitur yang berbeda sebagai perbandingan tingkat akurasi pengklasifikasian.

DAFTAR PUSTAKA

- [1] W. Anggraini, "Keanekaragaman Hayati Dalam Menunjang Perekonomian Masyarakat Kabupaten Oku Timur," *J. Aktual*, vol. 16, no. 2, p. 99, 2018, doi: 10.47232/aktual.v16i2.24.
- [2] P. T. Violetta, "KLHK: Luas tutupan hutan Indonesia capai 95,6 juta hektare," *www.antaraneews.com*, 2021. <https://www.antaraneews.com/berita/2588981/klhk-luas-tutupan-hutan-indonesia-capai-956-juta-hektare> (accessed Mar. 08, 2022).
- [3] A. E. Prasetyo, I. Lubis, and A. Budiman, "Prosiding Pemodelan 3D Virtual Reality Pada Tumbuhan Gymnospermae dan Angiospermae Sebagai Media Pembelajaran," *Semin. Nas. ...*, 2021, [Online]. Available: <http://prosiding.snastikom.com/index.php/SNASTIKOM2020/article/view/104>.
- [4] J. Safitri, P. Meilina, and S. Nurbaya Ambo, "Implementasi Augmented Reality Sebagai Pembelajaran Pertumbuhan Tanaman Dikotil Dan Monokotil Untuk Sekolah Dasar," *J. Sist. Informasi, Teknol. Inform. dan Komput.*, vol. 9, no. 1, pp. 32–38, 2018, doi: 10.24853/justit.9.1.32-38.
- [5] E. Budianita, T. Ulfadhiani, and F. Yanto, "Implementasi Algoritma Canny Dan Backpropagation Untuk Mengklasifikasi Jenis Tanaman Mangga," *Semin. Nas. Teknol. Informasi, Komun. dan Ind.*, vol. 11, pp. 13–21, 2019.
- [6] D. Syahid, J. Jumadi, and D. Nursantika, "Sistem Klasifikasi Jenis Tanaman Hias Daun Philodendron Menggunakan Metode K-Nearest Neighbor (KNN) Berdasarkan Nilai Hue, Saturation, Value (HSV)," *J. Online Inform.*, vol. 1, no. 1, p. 20, Jun. 2016, doi: 10.15575/join.v1i1.6.
- [7] S. A. Rosiva Srg, M. Zarlis, and W. Wanayumini, "Identifikasi Citra Daun dengan GLCM (Gray Level Co-Occurrence) dan K-NN (K-Nearest Neighbor)," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 2, pp. 477–488, 2022, doi: 10.30812/matrik.v21i2.1572.
- [8] D. K. Indahsari and Y. I. Kurniawan, "APLIKASI PREDIKSI USIA KELAHIRAN DENGAN METODE K-NEAREST NEIGHBOR," *J. Kebidanan*, vol. 11, no. 01, p. 1, Jul. 2019, doi: 10.35872/jurkeb.v11i01.335.
- [9] R. Perangin-angin and E. J. G. Harianja, "Comparison detection edge lines algoritma canny dan sobel," *J. TIMES*, vol. VIII, no. 2, pp. 35–42, 2019.
- [10] B. Sinaga, J. Manurung, M. H. Silalahi, and S. Ramen, "Deteksi Tepi Citra Dengan Metode Laplacian of Gaussian Dan Metode Canny," *J. Sains Komput. Inform.*, vol. 5, no. 2, pp. 1066–1084, 2021, doi: 10.30645/j-sakti.v5i2.401.
- [11] W. Supriyatn, "Perbandingan Metode Sobel, Prewitt, Robert dan Canny pada Deteksi Tepi Objek Bergerak," *Ilk. J. Ilm.*, vol. 12, no. 2, pp. 112–120, 2020, doi: 10.33096/ilkom.v12i2.541.112-120.
- [12] K. H. Mahmud, Adiwijaya, and S. Al Faraby, "Prosiding Klasifikasi Citra Multi-Kelas Menggunakan Convolutional Neural Network," *e-Proceeding Eng.*, vol. 6, no. 1, pp. 2127–2136, 2019.
- [13] S. A. Hantoush Alrubaie and A. H. Hameed, "Dynamic Weights Equations for Converting Grayscale Image to RGB Image," *J. Univ. BABYLON Pure Appl. Sci.*, vol. 26, no. 8, pp. 122–129, 2018, doi: 10.29196/jubpas.v26i8.1677.
- [14] A. Kurniasari and Jalinus, "Pendeteksian Tingkat Kepadatan Jalan Menggunakan Metode Canny Edge Detection," *J. Ilm. Teknol. dan Rekayasa*, vol. 25, no. 3, pp. 239–248, 2020, doi: 10.35760/tr.2020.v25i3.3419.
- [15] M. A. Masril, Yuhandri, and Jufriadif Na'am, "Analisis Perbandingan Perbaikan Kualitas Citra Pada Motif Batik Dengan Konsep Deteksi Tepi Robert, Sobel, Canny Menggunakan Metode Morfologi," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 1, pp. 36–41, 2019, doi: 10.29207/resti.v3i1.821.
- [16] A. Setiawan and K. M. Suryaningrum, "Optical Character Recognition Jepang Menggunakan Matriks Populasi Piksel Dan L1-Metric," *J. Ilm. Teknol. Infomasi Terap.*, vol. 4, no. 1, pp. 62–72, 2017, doi: 10.33197/jitter.vol4.iss1.2017.151.
- [17] M. Z. Luhing and K. M. Suryaningrum, "Pengenalan karakter huruf Rusia dengan algoritma perceptron," *Processor*, vol. 13, no. 1, pp. 1160–1172, 2018.
- [18] A. Amalia, A. Zaidiah, and I. N. Isnainiyah, "Prediksi Kualitas Udara Menggunakan Algoritma K- Nearest Neighbor," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 7, no. 2, pp. 496–507, 2022, doi: 10.29100/jipi.v7i2.2843.
- [19] T. Harlina and E. Handayani, "Klasifikasi Motif Batik Banyuwangi Menggunakan Metode K-Nearest Neighbor (K-NN) Berbasis Android," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 7, no. 1, pp. 82–96, Feb. 2022, doi: 10.29100/jipi.v7i1.2411.
- [20] A. Alfani W.P.R., F. Rozi, and F. Sukmana, "PREDIKSI PENJUALAN PRODUK UNILEVER MENGGUNAKAN METODE K-NEAREST NEIGHBOR," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 6, no. 1, pp. 155–160, Jun. 2021, doi: 10.29100/jipi.v6i1.1910.