

USULAN TUGAS AKHIR

**ANALISIS PENGENALAN POLA DAUN BERDASARKAN
FITUR CANNY EDGE DETECTION DAN FITUR GLCM
MENGUNAKAN METODE KLASIFIKASI K-NEAREST
NEIGHBOR(K-NN)**



Oleh:
Azizah Arif Paturrahman
F1D 016 013

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MATARAM
2020

USULAN TUGAS AKHIR

ANALISIS PENGENALAN POLA DAUN BERDASARKAN FITUR CANNY EDGE DETECTION DAN FITUR GLCM MENGGUNAKAN METODE KLASIFIKASI K-NEAREST NEIGHBOR(K-NN)

Telah diperiksa dan disetujui oleh Tim Pembimbing:

1. Pembimbing Utama



Fitri Bimantoro, S.T., M.Kom.
NIP. 19860622 201504 1 002

Tanggal: 27 April 2020

2. Pembimbing Pendamping



Prof. Dr. Eng. I Gede Pasek Suta Wijaya, ST., MT.
NIP. 19731130 200003 1 001

Tanggal: 27 April 2020

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Teknik
Universitas Mataram



Prof. Dr. Eng. I Gede Pasek Suta Wijaya, ST., MT.
NIP: 19731130 200003 1 001

USULAN TUGAS AKHIR
ANALISIS PENGENALAN POLA DAUN BERDASARKAN FITUR
CANNY EDGE DETECTION DAN FITUR GLCM MENGGUNAKAN
METODE KLASIFIKASI K-NEAREST NEIGHBOR(K-NN)

Oleh:

Azizah Arif Paturrahman
F1D 016 013

Susunan Tim Penguji

1. Penguji I



Gibran Satya Nugraha, S.Kom., M.Eng.
NIP: 19920323 201903 1 012

Tanggal: 18 April 2020



2. Penguji II



Ramaditia Dwiyanaputra, S.Kom., M.Eng.
NIP: -

Tanggal: 17 April 2020

3. Penguji III

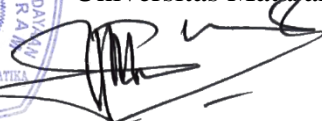


Moh. Ali Albar, S.Kom., M.Eng.
NIP: 19831125 201504 1 002

Tanggal: 25 April 2020

Mataram, 27 April 2020

Ketua Program Studi Teknik Informatika
Fakultas Teknik
Universitas Mataram



Prof. Dr. Eng. I Gede Pasek Suta Wijaya, S.T., MT.
NIP: 19731130 200003 1 001

DAFTAR ISI

BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	5
2.1 Tinjauan Pustaka.....	5
2.2 Dasar Teori	8
2.2.1 Citra Digital	8
2.2.2 Pengolahan Citra.....	8
2.2.3 Pengenalan Pola	8
2.2.4 Deteksi Tepi	10
2.2.5 Ekstraksi Fitur <i>Canny Edge Detection</i>	12
2.2.6 Ekstraksi Fitur <i>Gray Level Co-Occurrence Matrix (GLCM)</i>	14
2.2.7 <i>K-Nearest Neighbor (KNN)</i>	15
2.2.8 Daun.....	16
BAB III METODOLOGI PENELITIAN.....	5
3.1 Alat dan Bahan Penelitian	17
3.2 Proses Penelitian.....	17
3.3 Perancangan Sistem.....	17
3.3.1 <i>Preprocessing</i>	18
3.3.2 Ekstraksi Fitur.....	19
3.3.3 Klasifikasi	21

3.4 Skenario Pengujian.....	22
DAFTAR PUSTAKA.....	34

DAFTAR GAMBAR

Gambar 2.1 Proses Deteksi Tepi	10
Gambar 2.2 <i>Sample</i> citra daun	16
Gambar 3.1 Sample Data Citra Daun[2]	18
Gambar 3.2 Digram alir pembuatan sistem.....	18
Gambar 3.3 Diagram alur penelitian.....	19
Gambar 3.4 Tahapan <i>Preprocessing</i>	20
Gambar 3.5 Citra hasil <i>praprocessing</i>	21
Gambar 3.6 Citra hasil deteksi tepi <i>canny</i>	22
Gambar 3.7 Contoh penerapan filter <i>Gaussian</i>	23
Gambar 3.8 Contoh matriks dengan kernel <i>gx</i>	23
Gambar 3.9 Hasil Konvolusi dari kernel <i>gx</i>	23
Gambar 3.10 Contoh matriks dengan kernel <i>gy</i>	23
Gambar 3.11 Hasil Konvolusi dari kernel <i>gy</i>	24
Gambar 3.12 Hasil perhitungan matriks arah tepi	24
Gambar 3.13 Hasil matriks Gradien.....	24
Gambar 3.14 Hasil matriks <i>Non Maximus Supression</i>	25
Gambar 3.15 Hasil matriks <i>thresholding</i>	25
Gambar 3.16 Hasil matriks <i>canny</i>	25
Gambar 3.17 Citra asli <i>grayscale</i>	27
Gambar 3.18 Komposisi piksel.....	27
Gambar 3.19 Jumlah pasangan piksel dengan dengan sudut 90^0	27
Gambar 3.20 Matriks GLCM yang telah dinormalisasi.....	27

DAFTAR TABEL

Tabel 3.1 Nilai 4 fitur pada citra 2 spesies	29
Tabel 3.2 Menentukan nilai <i>Cosine similarity</i>	30
Tabel 3.3 Menentukan yang termasuk 3-NN	31
Tabel 3.4 Menentukan kategori spesies yang termasuk 3-NN.....	31
Tabel 3.5 Skenario Uji Akurasi dengan 5 - <i>fold cross validation</i> 1	32
Tabel 3.6 Skenario Uji Akurasi dengan 5 - <i>fold cross validation</i> 2	32

ABSTRAK

Daun merupakan salah satu ciri tumbuhan yang mudah diamati dan cukup representatif sehingga bisa dijadikan obyek untuk ekstraksi fitur ciri daun dan daun merupakan bagian tanaman yang sering digunakan dalam proses klasifikasi tanaman melalui pengamatan pada ciri-ciri fisik pada daun tersebut. Proses klasifikasi citra daun dapat dilakukan dengan cara mengidentifikasi citra bentuk daun yang dapat dilakukan dengan cara pengenalan pola daun dengan mengenali karakteristik struktural daun seperti bentuk dan tekstur daun tersebut. Klasifikasi citra daun ini berdasarkan fitur *canny edge detection* dan *Gray-Level Co-occurrence Matrix* (GLCM) menggunakan metode *K-Nearest Neighbor* (KKN) dengan jumlah data sebanyak 350 citra daun dengan 7 spesies yang berbeda yang diperoleh dari data penelitian sebelumnya.

Kata kunci : daun, *canny edge detection*, *gray-level co-occurrence Matrix* (GLCM), *K-nearest neighbor* (KKN).

BAB I

PENDAHULUAN

1.1 Latar Belakang

Ilmu tentang tumbuhan mengalami kemajuan yang pesat, bidang pengetahuan yang sebelumnya hanya merupakan cabang ilmu tumbuhan saja, sekarang telah menjadi bidang ilmu yang berdiri sendiri. Salah satunya adalah morfologi tumbuhan yang mempelajari bentuk dan susunan tubuh tumbuhan [1].

Determinasi tumbuhan merupakan proses menentukan pengklasifikasian jenis tumbuhan berdasarkan ciri morfologi, fisiologi maupun anatominya. Dalam penelitian ini dilakukan klasifikasi tumbuhan berdasarkan bentuk morfologi daun. Jumlah aneka flora dan variasinya yang semakin banyak menyebabkan kemiripan bentuk yang tinggi dan terkadang ditemukan kesulitan dalam pengambilan data daun secara utuh, terutama untuk daun yang memiliki skala besar[2] dengan ukuran yang berbeda-beda antara daun yang satu dengan daun yang lainnya. Oleh karena itu tekstur juga dibutuhkan untuk membantu proses klasifikasi daun terutama untuk daun yang memiliki kemiripan yang tinggi. Klasifikasi otomatis menggunakan sistem komputerisasi tentunya dapat sangat membantu masyarakat luas terutama peneliti di bidang ahli tanaman herbal, pertanian, perkebunan, dokter dan lain sebagainya[3].

Penelitian berbasis warna daun masih memiliki kendala, sebab sebagian besar daun memiliki warna dominan hijau. Oleh karena itu tekstur dan bentuk merupakan fitur daun yang paling sering diteliti. Penelitian tentang pengenalan pola daun sebelumnya telah dilakukan beberapa kali dengan penelitian daun berdasarkan bentuk dan tekstur yaitu penelitian yang menggunakan KNN yang berfokus pada penggabungan median filter dan erosi citra yang digunakan untuk memperbaiki proses fitur. Sedangkan untuk ekstraksi fitur menggunakan metode invarian moment[1], penelitian tentang klasifikasi berdasarkan fitur bentuk, warna dan tekstur daun menggunakan PNN (*Probability Neural Network*), penelitian

dengan menggunakan *Support Vector Machine* (SVM) dan melakukan kombinasi Metode filter *Gabor* dan matriks *Co-Occurrence* untuk menghasilkan fitur yang paling representatif untuk klasifikasi daun[4], kemudian penelitian dengan menggunakan metode *Local Binary Pattern* dan *Canny* melakukan klasifikasi tumbuhan dengan domain tekstur dan bentuk[2], serta penelitian dengan menggunakan fitur berbasis fraktal dari koefisien wavelet untuk klasifikasi citra daun dengan metode SVM[3].

Penggabungan fitur bentuk dan tekstur menghasilkan akurasi yang lebih baik daripada penggunaan fitur bentuk dan tekstur yang terpisah. Maka, pada penelitian ini melakukan penggabungan ekstraksi fitur bentuk menggunakan metode *Canny* dan ekstraksi fitur tekstur menggunakan metode GLCM (*Gray-Level Co-occurrence Matrix*) dengan yang merepresentasikan keunikan tiap daun kemudian digunakan sebagai masukan pada proses klasifikasi dengan menggunakan metode K-NN. *Dataset* yang digunakan yaitu citra daun flavia yang dataset yang berasal dari peneliti lain yang umum dipakai dalam penelitian identifikasi daun.

Metode klasifikasi *K-Nearest Neighbor* (KNN) melakukan klasifikasi terhadap objek berdasarkan data latih terdekat. Berdasarkan penelitian-penelitian yang telah dilakukan sebelumnya diantaranya yaitu penelitian tentang pengenalan jenis pempek menggunakan metode *K-Nearest Neighbors* memiliki rata-rata tingkat akurasi 30%-100% [5]. Selain itu, pada penelitian tentang pengenalan motif batik memiliki tingkat akurasi 66,67-100%[6].

Oleh karena itu penulis mengajukan sebuah penelitian untuk perancangan sebuah model analisis pengenalan pola daun berdasarkan fitur *Canny* dan fitur GLCM (*Gray-Level Co-occurrence Matrix*) menggunakan metode klasifikasi *K-Nearest Neighbors* (K-NN), keluaran dari penelitian ini diharapkan dapat mengidentifikasi jenis daun tersebut berdasarkan bentuk dan tekstur daun serta dapat mengetahui tingkat keakurasian metode dengan melakukan penggabungan

metode ekstraksi fitur *Canny* dan fitur GLCM (*Gray-Level Co-occurrence Matrix*) menggunakan metode klasifikasi *K-Nearest Neighbors* (K-NN).

1.2 Rumusan Masalah

Bagaimana performa dari metode *Canny Edge Detection*, GLCM (*Gray-Level Co-occurrence Matrix*) dan *K-Nearest Neighbors* (K-NN) untuk mengidentifikasi jenis daun berdasarkan bentuk dan tekstur daun tersebut ?

1.3 Batasan Masalah

Penelitian ini memiliki batasan-batasan masalah untuk memberikan lingkup penelitian agar lebih terfokus ketika pengerjaan. Adapun batasan masalah yang diberikan adalah sebagai berikut.

1. Citra daun yang digunakan adalah citra daun yang berwarna.
2. Citra daun yang dideteksi adalah hasil pemotretan dari depan (tampak depan).
3. Citra daun yang akan di proses tidak memiliki kerusakan (gangguan lainnya).

1.4 Tujuan

Mengetahui performa dari metode *Canny Edge Detection*, GLCM (*Gray-Level Co-occurrence Matrix*) dan *K-Nearest Neighbors* (K-NN).

1.5 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut :

1. Dapat mengetahui performa dari metode yang digunakan untuk pengenalan pola.
2. Dapat menjadi referensi untuk melakukan penerapan dalam hal pengenalan pola.
3. Model yang dihasilkan dari penelitian ini dapat digunakan untuk pembuatan sistem pengenalan pola (*pattern recognition*).
4. Dapat menjadi media pembelajaran untuk mengidentifikasi jenis daun berdasarkan ciri bentuk dan tekstur.
5. Hasil penelitian dapat memberikan masukan bagi peneliti, masyarakat, lembaga terkait dan pengembangan ilmu pengetahuan dan teknologi serta

diharapkan dapat menjadi bahan pertimbangan sebuah kebijakan maupun keputusan.

1.6 Sistematika Penulisan

Sistematika penulisan dari penelitian ini disajikan dalam beberapa bab antara lain sebagai berikut.

1. Bab I Pendahuluan

Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, serta sistematika penulisan laporan tugas akhir.

2. Bab II Tinjauan Pustaka dan Dasar Teori

Bab ini membahas tentang penelitian-penelitian terdahulu yang mengimplementasikan metode *Canny Edge Detection*, GLCM (*Gray-Level Co-occurrence Matrix*) dan *K-Nearest Neighbors* (K-NN) serta teori-teori sebagai referensi penulis ketika melakukan penelitian.

3. Bab III Metodologi Penelitian

Bab ini membahas tentang metodologi yang digunakan untuk pengenalan pola daun dengan metode *Canny Edge Detection* dan GLCM (*Gray-Level Co-occurrence Matrix*) menggunakan metode klasifikasi *K-Nearest Neighbors* (K-NN).

4. Bab IV Analisis dan Perancangan

Pada bab ini merupakan pembahasan tentang analisis perangkat lunak, meliputi analisis, analisis masalah, analisis metode, analisis kebutuhan sistem, serta perancangan sistem yang terdiri dari perancangan diagram alir (*flowchart*).

5. Bab V Implementasi dan Pengujian Metode

Bab ini membahas implementasi yang dilakukan terhadap pengolahan citra digital pada citra di *desktop* menggunakan *python* dengan metode *Canny Edge Detection* dan GLCM (*Gray-Level Co-occurrence Matrix*) menggunakan metode klasifikasi *K-Nearest Neighbors* (K-NN).

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Sebagai tinjauan pustaka, berbagai penelitian mengenai pengenalan pola daun telah dilakukan sebelumnya. Perbedaan paling menonjol dari setiap penelitian terletak pada *dataset* dan fitur-fitur yang digunakan. Perbedaan lain juga terletak pada metode klasifikasi yang digunakan. Penelitian-penelitian tersebut akan digunakan sebagai acuan dan pengetahuan untuk melaksanakan penelitian ini.

Penelitian tentang pengenalan pola daun sebelumnya telah dilakukan beberapa kali. Penelitian-penelitian yang dimaksud antara lain penelitian tentang Klasifikasi Tanaman berdasarkan Fitur Bentuk dan Tekstur pada Daun menggunakan *Decision Tree*, pada penelitian ini fitur yang digunakan untuk pengenalan pola daun adalah fitur bentuk dan tekstur. Fitur bentuk dilakukan melalui fitur *global*, yaitu *aspect ratio*, *rectangularity*, *convex area ratio*, *circularity* sedangkan ekstraksi fitur tekstur dilakukan sesuai *deskriptor lokal*, yaitu kontras, *correlation*, *energy*, *homogeneity*, *maximum probability*, dan *entropy* [7]. Pada penelitian tentang Klasifikasi Daun Dengan Perbaikan Fitur Citra Menggunakan Metode *K-Nearest Neighbor*, yang berfokus pada penggabungan median filter dan erosi citra yang digunakan untuk memperbaiki proses fitur. Sedangkan untuk ekstraksi fitur menggunakan metode *invariant moment* [1]. Penelitian tentang Fitur Berbasis Fraktal Dari Koefisien Wavelet Untuk Klasifikasi Citra Daun, pada penelitian ini penggunaan metode praproses dan ekstraksi wavelet pada ekstraksi fitur citra daun. Ekstraksi fitur menggunakan teknik perhitungan statistika sederhana pada koefisien wavelet sehingga komputasi menjadi lebih ringan. Hasil ekstraksi fitur citra daun akan menjadi data masukan untuk sistem klasifikasi *Support Vector Machine* (SVM)[3]. Penelitian tentang Analisis Citra Daun Berdasarkan Fitur *Local Binary Pattern* Dan Fitur *Canny Edge Detection* Menggunakan Metode Klasifikasi *K-Nearest Neighbor* (K-NN), pada penelitian ini

hasil menunjukkan bahwa dari kedua ekstraksi fitur yang digunakan, fitur *Local binary pattern* mendapatkan akurasi sebesar 94.28%, sedangkan fitur *Canny edge detection* mendapatkan akurasi tertinggi yaitu sebesar 98.85%. Serta dilakukannya penggabungan kedua fitur menghasilkan hasil akurasi yang meningkat yaitu sebesar 99.42%. Artinya dengan dilakukannya pengujian penggabungan kedua fitur mendapatkan hasil akurasi yang lebih baik, dibandingkan klasifikasi dengan ekstraksi fitur yang terpisah[2]. Penelitian lainnya yaitu Klasifikasi Citra Daun dengan Metode *Gabor Co-Occurrence*, pada penelitian ini menggabungkan metode filter *Gabor* dengan matrik *Co-occurrence* untuk menghasilkan klasifikasi daun yang representatif menggunakan metode SVM menghasilkan akurasi sebesar 89.83%[4].

Penelitian tentang pengenalan pola menggunakan metode deteksi tepi *Canny* sebelumnya telah dilakukan beberapa kali. Penelitian-penelitian yang dimaksud antara lain penelitian tentang Analisis Perbandingan Metode *Prewitt* dan *Canny* untuk Identifikasi Ikan Air Tawar dengan menggabungkan kedua metode tersebut dapat memperjelas konten dari objek ikan yang akan diidentifikasi [8], pada penelitian lainnya tentang Implementasi Metode *Canny* Untuk Deteksi Tepi Mutu Daun Tembakau memperoleh hasil akurasi 90% menggunakan metode *Canny* [9], penelitian tentang Pengubahan Data *Image* Ikan Air Tawar ke Data Vektor menggunakan *Edge Detection* Metode *Canny* memperoleh hasil deteksi tepi optimal dengan menggunakan metode *Canny*[8], dan penelitian tentang Implementasi Algoritma *Canny* dalam Pengenalan Wajah menggunakan Antarmuka GUI Matlab memperoleh hasil deteksi tepi wajah yang jelas [10].

Penelitian tentang pengenalan pola menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM) sebelumnya telah dilakukan beberapa kali. Penelitian-penelitian yang telah dilakukan sebelumnya diantaranya yaitu penelitian tentang Simulasi dan Analisis Pengenalan Citra Daging Sapi dan Daging Babi dengan Metode GLCM sudut 0° , 45° , 90° dan 135° dengan metode klasifikasi *Fuzzy Logic*

dan menghasilkan tingkat akurasi yang tinggi pada sudut 90° sebesar 98.5% dibandingkan dengan sudut-sudut yang lainnya [11]. Selain itu, terdapat penelitian tentang Identifikasi Kematangan Buah Apel Dengan *Gray Level Co-Occurrence Matrix* (GLCM) yang menghasilkan tingkat akurasi sebesar 95% [12] serta penelitian tentang Klasifikasi Penyakit Daun Padi Berdasarkan Hasil Ekstraksi Fitur GLCM Interval 4 Sudut menghasilkan tingkat akurasi sebesar 80% [13].

Penelitian tentang pengenalan pola menggunakan metode klasifikasi *K-Nearest Neighbor* sebelumnya telah dilakukan beberapa kali. Penelitian-penelitian yang dimaksud antara lain penelitian tentang Pengenalan Motif Batik Menggunakan Deteksi Tepi *Canny* dan *K-Nearest Neighbor* memperoleh tingkat akurasi 100% pada penggunaan data *testing* yang sama dengan data *training*, sedangkan penggunaan data *training* yang berbeda dengan data *testing* diperoleh akurasi tertinggi 66,67%. Kedua akurasi tersebut diperoleh dengan menggunakan *lower threshold* = 0.010 dan *upper threshold* = 0.115 dan menggunakan $k=1$ [6]. Penelitian tentang Klasifikasi Daun Dengan Perbaikan Fitur Citra Menggunakan Metode *K-Nearest Neighbor* memperoleh tingkat akurasi 86,67% [1], penelitian lainnya tentang Implementasi Metode *K-Nearest Neighbor* pada Pengenalan Pola Tekstur Citra Saliva untuk Deteksi Ovulasi memperoleh tingkat akurasi 93,3% untuk klasifikasi *infertile*, 100% untuk klasifikasi *intermediate*, dan 93,3% untuk klasifikasi *fertile* pada 15 citra uji [14], dan penelitian tentang Ekstraksi Fitur Histogram Guna Klasifikasi Jenis Citra Kayu Menggunakan Algoritma *K-Nearest Neighbor* memperoleh tingkat akurasi 91% dengan menggunakan 150 citra *training* dan 100 citra uji dan pengelompokan jenis kayu dengan nilai $k=3$ [15] serta penelitian tentang Identifikasi Kupu-Kupu Menggunakan Ekstraksi Fitur Deteksi Tepi (*Edge Detection*) dan Klasifikasi *K-Nearest Neighbor* (KNN) menghasilkan tingkat akurasi 80% [16].

Berdasarkan beberapa penelitian yang menjadi tinjauan pustaka tersebut, maka didapatkan kesimpulan bahwa pengenalan pola menggunakan metode deteksi

tepi *Canny* untuk ekstraksi fitur bentuk lebih optimal dibandingkan dengan metode *Prewitt*, *Sobel* dan *Laplace* dan ekstraksi fitur tekstur menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM) serta menggunakan metode klasifikasi *K-Nearest Neighbor* dapat menghasilkan akurasi hampir mencapai 100%.

Berdasarkan kesimpulan yang telah diuraikan, dapat diketahui bahwa metode *Canny Edge Detection*, *Gray Level Co-occurrence Matrix* (GLCM) dan metode klasifikasi *K-Nearest Neighbor* dapat memperoleh hasil yang optimal untuk pendeteksian tepi citra dan pengklasifikasian citra. Oleh karena itu, penulis bermaksud untuk menggunakan metode ini untuk analisis pengenalan pola daun.

2.2 Dasar Teori

2.2.1 Citra Digital

Citra digital adalah citra kontinu yang diubah dalam bentuk diskrit, baik koordinat ruang maupun intensitas cahayanya. Citra digital dapat dinyatakan dalam matrik dua dimensi $f(x,y)$ dimana 'x' dan 'y' merupakan koordinat *pixel* dalam matrik dan 'f' merupakan derajat intensitas *pixel* tersebut. Citra digital berbentuk matrik dengan ukuran M x N akan tersusun sebagai berikut [17].

$$f_B(x,y) = O_{titik}\{f_A(x,y)\} \quad (2-1)$$

2.2.2 Pengolahan Citra

Secara umum, pengolahan citra bertujuan untuk memperbaiki kualitas citra agar mudah diinterpretasikan oleh manusia atau komputer. Pengolahan citra mentransformasikan citra menjadi citra lain, artinya masukannya berupa citra dan keluarannya juga berupa citra tetapi kualitasnya lebih baik dari citra masukan [14]. Pengolahan citra digital merupakan proses yang bertujuan untuk menganalisis citra dengan bantuan komputer. Filter *smoothing* digunakan untuk mengaburkan dan mengurangi *noise* pada tahap *preprocessing*[10].

2.2.3 Pengenalan Pola

Secara umum pengenalan pola (*pattern recognition*) adalah suatu ilmu untuk mengklasifikasi atau menggambarkan sesuatu berdasarkan pengukuran

kuantitatif fitur (ciri) atau sifat utama dari suatu objek. Pola sendiri adalah suatu entitas yang terdefinisi dan dapat diidentifikasi serta diberi nama. Pengenalan pola bertujuan menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Dengan kata lain, pengenalan pola membedakan suatu obyek dengan obyek lain [17]. Tahapan pengenalan pola sebagai berikut :

a. *Preprocessing*

Preprocessing bertujuan untuk meningkatkan kualitas suatu citra. Metode yang dilakukan pada *preprocessing* yaitu *scaling*, *segmentation*, dan *grayscale*. *Scaling* adalah merubah ukuran piksel suatu citra menjadi lebih besar atau lebih kecil untuk diambil pola dari suatu citra. *Segmentation* adalah tahap menghilangkan bagian-bagian citra (*background*) yang tidak diperlukan. *Grayscale* yaitu mengubah suatu citra yang memiliki 3 layer warna RGB (*Red Green Blue*) menjadi citra *grayscale* (keabu-abuan) [16].

b. *Feature Extraction*

Tujuan *feature extraction* adalah untuk mendapatkan informasi (ciri) pada suatu citra, ekstraksi fitur merupakan proses yang penting dalam mengklasifikasikan objek pada pengenalan pola [16]. Citra daun yang diklasifikasi dengan baik dapat digunakan untuk pengenalan jenis daun.

c. *Classification*

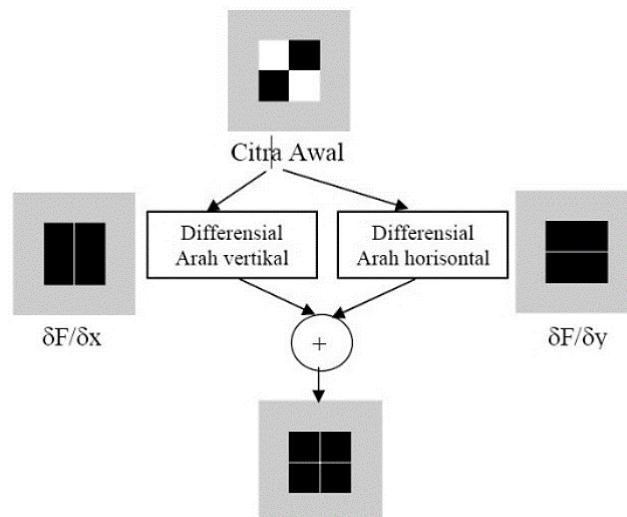
Classification adalah proses penemuan model yang membedakan antara kelas latih yang digunakan untuk memprediksi kelas uji yang belum diketahui kelasnya. Metode klasifikasi yang banyak digunakan yaitu *Rough Sets*, *Naive Bayes classifiers*, *Memory Based Reasoning*, *Neural Network*, *Classification Trees*, *K-Nearest Neighbor*, *Extreme Learning Machine*, dan *Support Vector Machines*. Proses ini dilakukan agar data atau citra dapat dikategorikan dalam suatu kelas tertentu yang telah ditentukan [16].

2.2.4 Deteksi Tepi

Deteksi tepi merupakan salah satu proses penting di dalam *image processing*. Tepi suatu *image* dapat dijadikan salah satu ciri atau fitur yang dapat digunakan dalam proses pengenalan (identifikasi dan *recognition*), klasifikasi dan penjabaran informasi dari suatu *image* [18]. Tepi obyek bermanfaat untuk segmentasi, registrasi dan identifikasi pada obyek. Titik-titik tepi dapat dikatakan sebagai tempat perubahan piksel pada kekasaran tingkat keabuan. Deteksi tepi adalah proses untuk menemukan perubahan intensitas berbeda dalam sebuah bidang citra [19]. Deteksian tepi suatu citra akan menghasilkan tepi-tepi dari objek citra, tujuannya antara lain :

- Untuk menandai bagian yang menjadi detail citra.
- Memperbaiki detail citra yang kabur.
- Adanya efek proses akuisisi citra.
- Mengubah citra 2D menjadi bentuk kurva

Gambar 2.1 berikut ini, memperlihatkan bagaimana tepi pada sebuah gambar diperoleh.



Gambar 2.0.1 Proses Deteksi Tepi

Salah satu operator deteksi tepi adalah deteksi tepi *Canny*. Operator *Canny* merupakan salah satu metode deteksi tepi modern. Operator *canny*, yang dikemukakan oleh Jhon Canny pada tahun 1986, terkenal sebagai operator deteksi

tepi yang optimal [20]. Ada beberapa kriteria pendeteksi tepian paling optimum yang dapat dipenuhi oleh metode *Canny*:

- a. Mendeteksi dengan baik (kriteria deteksi). Kemampuan untuk meletakkan dan menandai semua tepi yang ada sesuai dengan pemilihan parameter-parameter konvolusi yang dilakukan. Sekaligus juga memberikan fleksibilitas yang sangat tinggi dalam hal menentukan tingkat deteksi ketebalan tepi sesuai yang diinginkan.
- b. Melokalisasi dengan baik (kriteria lokalisasi). Dengan *Canny* dimungkinkan dihasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli.
- c. Respon yang jelas (kriteria respon). Hanya ada satu respon untuk tiap tepi. Sehingga mudah dideteksi dan tidak menimbulkan kerancuan pada pengolahan citra selanjutnya. Pemilihan parameter deteksi tepi *Canny* sangat mempengaruhi hasil dari tepian yang dihasilkan. Beberapa parameter tersebut adalah nilai standart deviasi *Gaussian* dan nilai ambang (*threshold*) [21].

Langkah-langkah dalam melakukan deteksi tepi *Canny Edge Detection* [16]:

1. Penghalusan citra (*smoothing*) menggunakan *Gaussian Filter*, memiliki persamaan matematika seperti pada rumus (2-2) :

$$G(i, j) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{(i-u)^2+(j-v)^2}{2\sigma^2}} \quad (2-2)$$

dimana :

$e = 2.71$ (konstanta euler)

σ = standar deviasi (sigma)

$\Pi = 3.14$ (pi)

2. Mendapatkan kekuatan tepi (*edge strength*) dengan menggunakan persamaan seperti pada rumus (2-3) :

$$G = \sqrt{|Gx|^2 + |Gy|^2} \quad (2-3)$$

- d. Menentukan arah tepi dengan menggunakan persamaan seperti pada rumus (2-4) :

$$\delta = \tan^{-1} \left(\frac{G_y}{G_x} \right) \quad (2-4)$$

- e. Garis tepi diperkecil dengan menerapkan *Non Maximus Suppression*
Penghilangan *non-maximum* dilakukan di sepanjang tepi dan menghilangkan piksel – piksel yang tidak dianggap sebagai tepi. Hanya nilai maksimum yang ditandai sebagai tepi. Sehingga didapatkan garis tepi yang lebih ramping [10].
- f. Binerisasi dengan menggunakan *hysteresis thresholding*
Tahap ini adalah klasifikasi tiap piksel apakah termasuk dalam kategori piksel tepi atau tidak dengan diterapkan dua buah *thresholding* yaitu *low thresholding* (T1) dan *high thresholding* (T2). Nilai yang kurang dari T1 akan diubah menjadi hitam (nilai 0) dan nilai yang lebih dari T2 diubah menjadi putih nilai 255) [10].

2.2.5 Ekstraksi Fitur

Metode *Canny* adalah metode yang memaksimalkan deteksi titik tepi yang benar dengan cara memaksimalkan *signal to noise* dan tepi yang terdeteksi harus sedekat mungkin dengan tepi yang nyata. Tujuan dari dilakukannya pendeteksian tepi yaitu untuk meningkatkan penampakan garis batas suatu daerah yang terdapat pada objek di dalam suatu citra[2]. Pada penelitian ini menggunakan empat jenis ekstraksi fitur bentuk, diantara adalah:

a. *Area*

Menghitung nilai skalar yang menentukan jumlah piksel aktual di wilayah tersebut. Yang membedakan adalah pola pikselnya berbeda-beda.

$$Area = \sum_{i=1}^n \sum_{j=1}^m f(i, j) \quad (2-5)$$

dimana : m = Jumlah baris piksel citra,

n = Jumlah kolom piksel citra ,

(i, j) = Koordinat spasial, dan

f(i, j) = 1 jika (i, j) adalah piksel objek dalam citra biner

b. *Perimeter*

Perbandingan antara keliling, panjang serta lebar daun merupakan salah satu perhitungan yang digunakan untuk ekstraksi fitur bentuk.

$$P = \frac{p}{Lp+Wp} \quad (2-6)$$

dimana : P = Rasio *Perimeter*,
 p = Keliling Daun,
 Lp = Panjang Daun, dan
 Wp = Lebar Daun

c. *Eccentricity*

Menghitung nilai skalar yang menentukan eksentrisitas elips, yang merupakan rasio jarak antara fokus elips dan panjang sumbu utamanya. Nilai pada eksentrisitas adalah antara 0 dan 1. Elips yang nilai eksentrisitasnya 0 merupakan sebuah lingkaran, sedangkan elips yang eksentrisitasnya 1 adalah segmen garis.

dengan persamaan :

$$e = \sqrt{1 - \frac{b^2}{a^2}} \quad (2-7)$$

dimana :

e = *Eccentricity*

a = *MajorAxisLength*

b = *MinorAxisLength*

d. *Matric*

Merupakan nilai perbandingan antara luas dan keliling objek. *Matric* memiliki rentang nilai antara 0 hingga 1. Objek yang memiliki bentuk memanjang atau mendekati bentuk garis lurus, nilai *metric*nya mendekati angka 0, sedangkan objek yang memiliki bentuk bulat atau lingkaran, nilai *metric*nya mendekati angka 1.

dengan persamaan :

$$M = \frac{4\pi \times A}{C^2} \quad (2-8)$$

dimana :

M = *Matric*

$A = Area$

$C = Circumference$

2.2.6 Ekstraksi Fitur - (GLCM)

Gray Level Co-occurrence Matrix (GLCM) adalah dengan mengubah citra RGB menjadi citra berskala keabuan. Langkah kedua adalah menciptakan matrik *co-occurrence* dan dilanjutkan dengan menentukan hubungan spasial antara piksel referensi dan piksel tetangga berdasarkan sudut θ dan jarak d . Langkah selanjutnya adalah menciptakan matrik simetris dengan menambahkan matrik *Co-occurrence* dengan matrik transpose-nya. Kemudian dilakukan normalisasi terhadap matrik simetris dengan menghitung probabilitas setiap elemen matrik. Langkah terakhir adalah dengan menghitung fitur GLCM. Setiap fitur dihitung dengan satu piksel jarak di empat arah, yaitu 0° , 45° , 90° , dan 135° untuk mendeteksi *co-occurrence* [11]. Pada penelitian ini menggunakan sudut 90° dengan fitur-fitur GLCM antara lain :

- a. *Angular Second Moment* (ASM) atau *Energy*

ASM merupakan ukuran homogenitas dari suatu citra.

$$ASM = \sum \sum (GLCM(i, j))^2 \quad (2-9)$$

- b. Kontras

Kontras merupakan ukuran keberadaan variasi aras keabuan piksel citra.

$$Kontras = \sum \sum |i - j|^2 GLCM(i, j) \quad (2-10)$$

- c. Entropi

Entropi menyatakan ukuran ketidakteraturan aras keabuan di dalam citra.

$$Entropi = - \sum \sum (GLCM(i, j)) \log (GLCM(i, j)) \quad (2-11)$$

d. *Homogeneity*

Homogeneity menyatakan ukuran kedekatan setiap elemen.

$$Homogeneity = \sum \sum \frac{GLCM(i,j)}{1+|i-j|} \quad (2-12)$$

2.2.7 K- (KNN)

Algoritma *K-Nearest Neighbor* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing – masing dimensi merepresentasikan fitur dari data. Algoritma *k- Nearest Neighbor* termasuk dalam golongan *supervised learning* yang bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola data yang sudah ada dengan data yang baru. Algoritma *k-Nearest Neighbor* bekerja berdasarkan klasifikasi ketetanggaan sebagai nilai prediksi dari sampel uji yang baru[1]. *Cosine similarity* digunakan untuk menghitung jarak antara dua vektor yang berfungsi menguji ukuran yang bisa digunakan sebagai interpretasi kedekatan jarak antara dua objek yang direpresentasikan dalam persamaan

$$\cos a = \frac{A.B}{||A||B|} = \frac{\sum_{i=1}^n A_i x B_j}{\sqrt{\sum_{i=1}^n (A_i)^2} x \sqrt{\sum_{i=1}^n (B_j)^2}} \quad (2-13)$$

dimana :

a =jarak uji ke data pembelajaran

A_i = data uji ke- i , dengan $i=1,2,\dots,n$

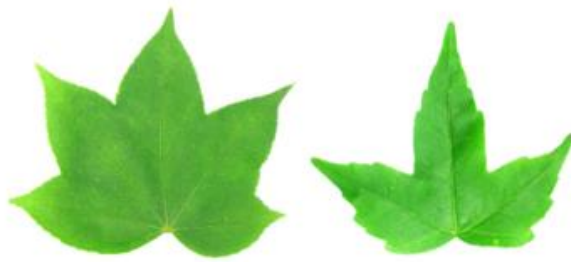
B_j =data uji ke- j , dengan $j=1,2,\dots,n$

Diperlukan perhitungan akurasi sebagai tolak ukur evaluasi dalam sistem. Pengukuran akurasi dapat menggunakan berbagai cara salah satunya menggunakan *recognition rate*. Persamaan *recognition rate* yang digunakan dapat dilihat pada rumus berikut:

$$Akurasi = \frac{\text{jumlah data benar}}{\text{jumlah data keseluruhan}} \times 100 \quad (2-14)$$

2.2.8 Daun

Daun merupakan bagian tanaman yang sering digunakan dalam proses klasifikasi tanaman melalui pengamatan pada ciri-ciri fisik pada daun tersebut. Bentuk tepi daun bisa digunakan untuk acuan klasifikasi daun. Proses klasifikasi citra daun dapat dilakukan dengan cara mengidentifikasi citra bentuk daun yang dapat dilakukan dengan cara pengenalan pola daun dengan mengenali karakteristik struktural daun seperti bentuk dan tekstur daun tersebut [2]. Spesies-spesies daun yang digunakan pada penelitian ini yaitu, spesies *Acer buergerianum*, *Aesculus chinensis*, *Cinnamomum japonicum* Sieb, *Kalopanax septemlobus*, *Phoebe nanmu* (Olive.) Gamble, *Pittosporum tobira* (Thunb.) Ait.f., *Prunus serrulata* Lindl. var. Gambar 2.2 berikut ini, memperlihatkan bagaimana bentuk citra daun yang akan digunakan pada penelitian ini.



Gambar 2.2 *Sample* citra daun

BAB III

METODOLOGI PENELITIAN

3.1 Alat dan Bahan Penelitian

Alat dan bahan pada penelitian yang dilakukan berupa *software* dan *hardware* serta data-data yang dibutuhkan selama kegiatan.

1. Alat Penelitian

Alat-alat yang akan digunakan dalam melakukan penelitian ini adalah sebagai berikut :

Processor Laptop : ACER E5-476G-34HM, Intel Core i3-6006U.

Sistem Informasi : Windows 10 64-bit.

Bahasa Pemrograman : Python.

2. Bahan Penelitian

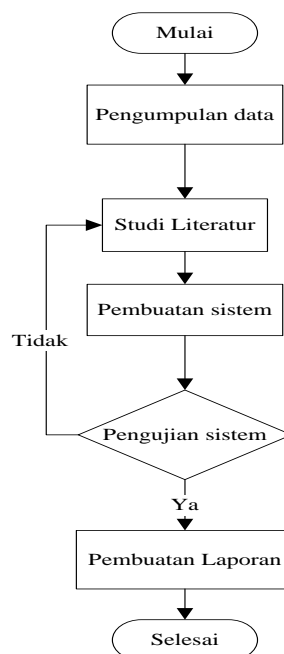
Pengumpulan data citra daun dilakukan dengan mengambil *dataset* citra daun Flavia dari sumber data citra daun yang terdapat di <https://sourceforge.net/projects/flavia/files/Leaf%20Image%20Dataset/1.0/Leaves.tar.bz2/download> dengan format file ".tar.bz2". *Dataset* citra daun Flavia yang terdiri dari 33 spesies citra daun dengan total jumlah data sebanyak 1.907 data citra daun, yang kemudian akan digunakan dalam penelitian ini sebanyak 7 spesies dengan jumlah data citra daun masing masing dari spesies sebanyak 50 data citra daun, dengan total keseluruhan data citra daun yang digunakan dalam penelitian ini sebanyak 350 data citra [2]. Pemilihan 7 spesies daun pada penelitian ini mengacu pada penelitian sebelumnya, dimana sudah terdapat nama spesies daun, namun hanya 7 spesies yang di ketahui, dikarenakan pada citra daun Flavia ini tidak terdapat nama-nama spesies daun tersebut.



Gambar 3.1 *Sample Data Citra Daun*[2]

3.2 Proses Penelitian

Guna mendukung berjalannya penelitian, proses penelitian ini dilakukan untuk mengetahui tahapan-tahapan atau alir pembuatan sistem yang akan dilakukan pada penelitian ini. Diagram alir pembuatan sistem dari mulai pengumpulan data hingga pembuatan laporan tertuang dalam Gambar 3.2.



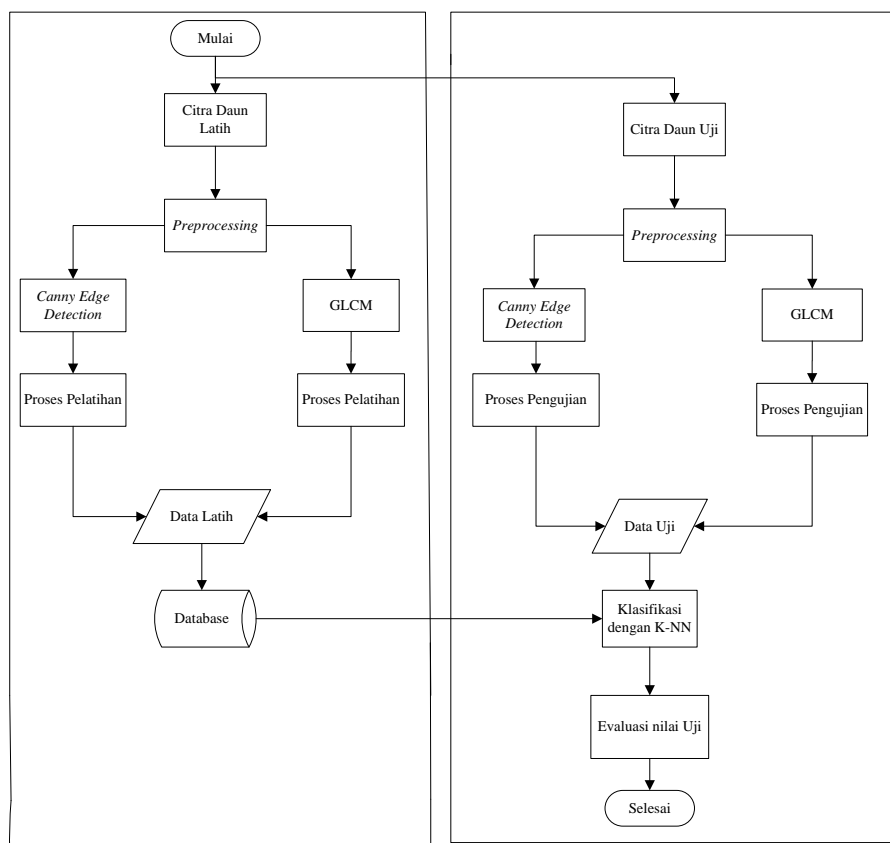
Gambar 3.2 Diagram alir pembuatan sistem

Berdasarkan Gambar 3.2, ada beberapa tahapan yang dilakukan dalam penelitian ini, pertama akan melakukan pengumpulan data citra daun selanjutnya melakukan studi literatur untuk menambah wawasan penulis mengenai apa yang akan dilakukan. Studi literatur ini dilakukan melalui referensi-referensi yang berasal dari jurnal, buku, dan penelitian-penelitian yang pernah dilakukan

sebelumnya untuk mempelajari cara membangun sistem sesuai dengan metode yang digunakan. Selanjutnya adalah tahap pembangunan sistem sesuai dengan rancangan yang telah dibuat. Tahap pengujian dilakukan untuk menguji apakah sistem sesuai dengan tujuan penelitian yaitu untuk mengetahui performa dari metode yang digunakan, apabila belum sesuai maka langkah selanjutnya akan kembali ke studi literatur. Setelah sistem berhasil dibangun maka tahap terakhir yakni pembuatan laporan.

3.3 Perancangan Sistem

Di dalam sub bab ini dijelaskan bagaimana sistem Analisis Pengenalan Pola Daun ini dirancang mulai dari tahapan pelatihan citra hingga sistem dapat mengetahui akurasi optimal dari citra latih dan citra uji. *Block diagram* tertuang dalam Gambar 3.3.

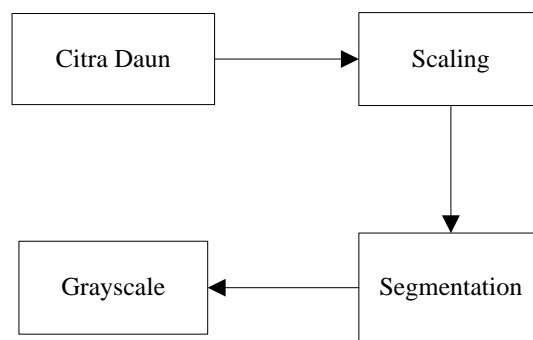


Gambar 3.3 Diagram alur penelitian

Berdasarkan Gambar 3.3, dapat dilihat bahwa sistem terbagi menjadi 2 proses yaitu proses pelatihan dan pengujian. Dimana proses pelatihan dan pengujian pada penelitian ini sama untuk tahap *preprocessing* dan ekstraksi fitur. Penjelasan diagram alur penelitian pada Gambar 3.3 adalah sebagai berikut:

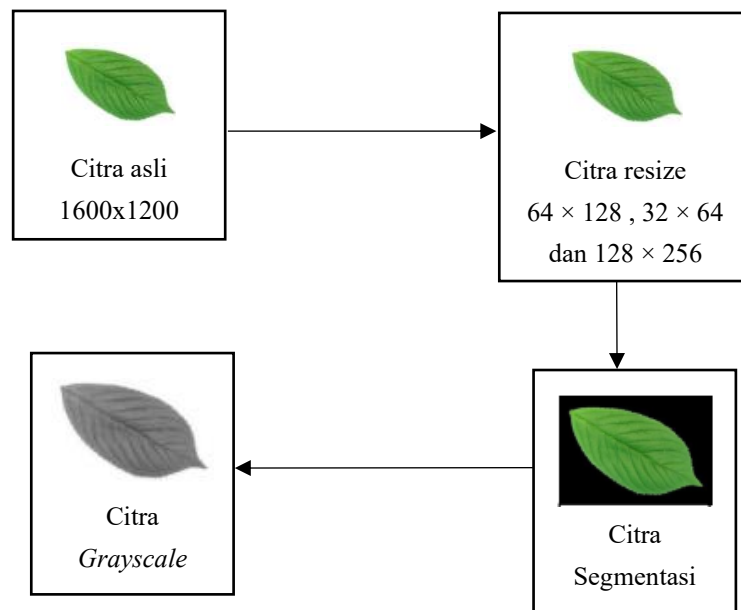
3.3.1

Preprocessing dilakukan dengan tujuan untuk mengolah data masukan sehingga meningkatkan kualitas citra dan dapat digunakan untuk proses ekstraksi fitur. Terdapat beberapa langkah yang dilakukan untuk *preprocessing* yaitu melakukan *scaling* dengan mengubah ukuran dari *dataset* asli citra daun Flavia, setelah melakukan *scaling* selanjutnya melakukan segmentasi yaitu menghilangkan bagian-bagian citra yang tidak diperlukan dan selanjutnya citra RGB diubah menjadi citra *grayscale*.



Gambar 3.4 Tahapan *Preprocessing*

Tahap awal dari *preprocessing* ini dengan mengubah ukuran citra asli dataset Flavia dengan ukuran awal 1600×1200 akan di *resize* menjadi 64×128 , 32×64 dan 128×256 [22]. Pembagian citra menjadi beberapa ukuran untuk menentukan resolusi (derajat rincian yang dapat dilihat) *special* yang diperoleh dan juga bertujuan untuk mempercepat proses komputasi menjadi lebih cepat. Berikut adalah tahapan *preprocessing* yang dapat dilihat pada Gambar 3.5.



Gambar 3.5 Citra hasil *praprocessing*

Setelah melalui tahap *preprocessing* selanjutnya melakukan ekstraksi fitur *canny edge detection* dengan tahapan melakukan *smoothing* dengan menggunakan *Gaussian filter*, *finding gradien*, *non-maximum suppression*, *double thresholding*, dan *edge tracking by hysteresis*. Setelah melakukan tahapan-tahapan *Canny edge detection* selanjutnya melakukan ekstraksi fitur bentuk dengan menghitung menghitung nilai *Area*, *Perimeter*, *Eccentricity* dan *Matric*. Sama halnya dengan *canny*, setelah *processing* dilakukan juga proses ekstraksi fitur tekstur dengan menggunakan *GLCM* dengan sudut 90^0 , pemilihan sudut ini karena memiliki tingkat akurasi yang tinggi dibandingkan sudut yang lainnya kemudian menghitung empat fitur tekstur yaitu *Contrast*, *Homogeneity*, *Energy*, dan *Entropy*. Setelah semua nilai fitur didapatkan maka akan dilakukan klasifikasi dengan menggunakan metode *KNN*.

3.3.2 Ekstraksi Fitur

1. Ekstraksi Fitur Bentuk *Canny Edge Detection*

Ekstraksi fitur dilakukan menggunakan metode *canny edge detection*

terhadap seluruh citra *dataset* daun Flavia. Pada tahap deteksi tepi ini merupakan langkah awal dari segmentasi citra untuk mendapatkan informasi dalam citra. Contoh citra disajikan pada Gambar 3.6.



Gambar 3.6 Citra hasil deteksi tepi *canny*

Berikut contoh perhitungan dari metode ini sebagai berikut :

a. *Filter Gaussian*

Untuk menghitung atau menentukan nilai-nilai dari setiap elemen dalam filter penghalus *Gaussian* yang akan dibentuk seperti Persamaan (2-2). Dalam hal ini, σ adalah standar deviasi dan piksel pada pusat (x, y) dengan bobot terbesar berupa 1. Sebagai contoh, bobot-bobotnya dapat diperoleh dengan contoh sebagai berikut :

$$G(0,0) = e^{-0} = 1$$

$$G(1,0) = G(0,1) = G(-1,0) = G(0,-1) = e^{-1/2} = 0,6065$$

$$G(1,1) = G(1,-1) = G(-1,1) = G(-1,-1) = e^{-1} = 0,3679$$

$$G(2,1) = G(1,2) = G(-2,1) = G(-1,-2) = e^{-5/2} = 0,0821$$

$$G(2,0) = G(0,2) = G(0,-2) = G(-2,0) = e^{-2} = 0,1353$$

$$G(2,2) = G(-2,-2) = G(-2,2) = G(2,-2) = e^{-4} = 0,0183$$

Langkah selanjutnya adalah melakukan proses *filtering* dengan menerapkan operasi konvolusi sebagai berikut :

0	0	1	1
0	2	1	1
0	2	2	2
2	2	3	3

Gambar 3.7 Contoh penerapan filter *Gaussian*

b. Menghitung Gradien

Menentukan gradien gambar yang telah diperhalus dengan menggunakan operator Sobel. Untuk melakukan proses perhitungan gradien dengan menerapkan proses konvolusi dengan persamaan yang akan dibentuk seperti Persamaan (2-4).

0	0	1	1
0	2	1	1
0	2	2	2
2	2	3	3

$$\text{kernel } gx = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

Gambar 3.8 Contoh matriks dengan kernel *gx*

0	0	1	1
0	5	-1	1
0	6	0	2
2	2	3	3

Gambar 3.9 Hasil Konvolusi dari kernel *gx*

0	0	1	1
0	2	1	1
0	2	2	2
2	2	3	3

$$\text{kernel } gy = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

Gambar 3.10 Contoh matriks dengan kernel *gy*

0	0	1	1
0	-7	5	1
0	-4	-6	2
2	2	3	3

Gambar 3.11 Hasil Konvolusi dari kernel gy

c. Mengitung sudut

Pada proses ini yaitu menghitung arah tepi yang ditentukan dengan menggunakan persamaan yang telah dibentuk pada Persamaan (2-5). Dengan hasil perhitungan arah tepi sebagai berikut :

45	45	45	45
45	45	88	45
45	23	0	45
45	45	45	45

Gambar 3.12 Hasil perhitungan matriks arah tepi

Setelah arah tepi diperoleh, maka perlu menghubungkan antara arah tepi dengan sebuah arah yang data dilacak dari citra, dengan pembagian warna yang berkisar antara :

$$0^0 = 0 - 22.5 \text{ dan } 157.5 - 180$$

$$45^0 = 22.5 - 67.5$$

$$90^0 = 67.5 - 112.5$$

$$135^0 = 112.5 - 157.5$$

45	45	45	45
45	45	90	45
45	45	0	45
45	45	45	45

Gambar 3.13 Hasil matriks Gradien

d. *Non Maximus Supression*

Pada tahap ini garis tepi diperkecil dengan menerapkan *Non Maximus Supression*, dimana dilakukan dengan menghilangkan piksel-piksel yang tidak dianggap sebagai tepi. Dengan hasil matriks *Non Maximus Supression* arah tepi sebagai berikut :

0	0	1	1
0	7	7	1
0	0	0	2
2	2	3	3

Gambar 3.14 Hasil matriks *Non Maximus Supression*

e. *Hysteresis Thresholding*

Pada tahap ini diterapkan dua buah *thresholding* yaitu *low thresholding* (T1) dan *high thresholding* (T2). Nilai yang kurang dari T1 akan diubah menjadi hitam (nilai 0) dan nilai yang lebih dari T2 diubah menjadi putih (nilai 1). Misalkan T1= 0 dan T2=255.

1	1	1	1
1	1	1	1
1	0	0	1
1	1	1	1

Gambar 3.15 Hasil matriks *thresholding*

Setelah melalui tahap *Canny edge detection* selanjutnya melakukan penghitungan empat ekstraksi fitur bentuk dengan hasil matriks *canny*, sebagai berikut :

1	1	1	1
1	1	1	1
1	0	0	1
1	1	1	1

Gambar 3.16 Hasil matriks *canny*

Fitur-fitur bentuk yang akan di dihitung diantaranya yaitu *Area*, *Perimeter*, *Eccentricity* dan *Matric*. Diantaranya sebagai berikut :

a. *Area*

Fitur Area ini berfungsi untuk menghitung luas suatu citra, dengan Persamaan (2-5) yaitu :

$$Area = \sum_{i=1}^n \sum_{j=1}^m f(i, j) = Area = 5 \times 5 = 25$$

b. *Perimeter*

Perbandingan antara keliling ,panjang serta lebar daun merupakan salah satu perhitungan yang digunakan untuk ekstraksi fitur bentuk, dengan Persamaan (2-6) yaitu :

$$P = \frac{p}{Lp+Wp} = P = \frac{20}{5+5} = 2$$

c. *Eccentricity*

Menghitung nilai skalar yang menentukan eksentrisitas elips. Elips yang nilai eksentrisitasnya 0 merupakan sebuah lingkaran, sedangkan elips yang eksentrisitasnya 1 adalah segmen garis, dengan Persamaan (2-7) yaitu :

$$e = \sqrt{1 - \frac{b^2}{a^2}} = \sqrt{1 - \frac{3^2}{3}} = 1$$

d. *Matric*

Menghitung nilai perbandingan antara luas dan keliling objek. Objek yang memiliki bentuk memanjang atau mendekati bentuk garis lurus, nilai metricnya mendekati angka 0, sedangkan objek yang memiliki bentuk bulat lingkaran, nilai metricnya mendekati angka 1, dengan persamaan (2-8) yaitu :

$$M = \frac{4\pi \times A}{C^2} = \frac{4 \times 3.14 \times 25}{20^2} \quad M = 0.58$$

2. Ekstraksi Fitur Tekstur *Gray Level Co-Occurrence Matrix* (GLCM)

Proses ekstraksi fitur dengan metode GLCM dibagi menjadi dua tahap, yaitu pembentukan matriks GLCM dan perhitungan nilai fitur GLCM sampai mencari nilai empat fitur.

a. Pembentukan matriks GLCM

Misalkan terdapat citra *grayscale* dengan matriks pada Gambar 3.17.

0	0	1	1
0	2	1	1
0	2	2	2
2	2	3	3

Gambar 3.17 Citra asli *grayscale*

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

Gambar 3.18 Komposisi piksel

4	0	2	0
0	4	2	0
2	2	4	2
0	0	2	0

Gambar 3.19 Jumlah pasangan piksel dengan dengan sudut 90^0

Matriks GLCM kemudian dinormalisasi agar jumlah seluruh elemennya sama dengan satu dengan rumus $M_{ij} = \frac{M_{ij}}{(\sum \sum (i,j))}$. Gambar 3.20 merupakan hasil normalisasi terhadap matriks B.

0.167	0	0.08	0
0	0,167	0.08	0
0.08	0.08	0.167	0.08
0	0	0.08	0

Gambar 3.20 Matriks GLCM yang telah dinormalisasi

b. Perhitungan nilai fitur GLCM

Matriks GLCM yang telah dinormalisasi pada tahap sebelumnya digunakan untuk menghitung 4 fitur GLCM yang terdiri atas *Energy*, *Entropy*, *Contrast*, dan *Homogeneity*, dari perhitungan kemudian akan menjadi masukkan pada tahap pelatihan ataupun klasifikasi. Berikut ini menunjukkan hasil perhitungan empat fitur GLCM pada sudut 90^0 .

1. *Energy*

Untuk menghitung nilai *Energy*, digunakan Persamaan (2-6).

$$\begin{aligned} \text{Energy} = & (0.167)^2 + (0)^2 + (0.08)^2 + (0)^2 + (0)^2 + (0.167)^2 + \\ & (0.08)^2 + (0)^2 + (0.08)^2 + (0.08)^2 + (0.167)^2 + (0.08)^2 + (0)^2 + \\ & (0)^2 + (0.08)^2 + (0)^2 \end{aligned}$$

$$\text{Energy} = 0.122$$

2. *Entropy*

Untuk menghitung nilai *Entropy*, digunakan Persamaan (2-7).

$$\begin{aligned} \text{Entropy} = & (-0.167 \times \log(0.167)) + (-0 \times \log(0)) + (-0.08 \times \\ & \log(0.08)) + (-0 \times \log(0)) + (-0 \times \log(0)) + (-0.167 \times \log(0.167)) + \\ & (-0.08 \times \log(0.08)) + (-0 \times \log(0)) + (-0.08 \times \log(0.08)) + (-0.08 \times \\ & \log(0.08)) + (-0.167 \times \log(0.167)) + (-0.08 \times \log(0.08)) + (-0 \times \\ & \log(0)) + (-0 \times \log(0)) + (-0.08 \times \log(0.08)) + (-0 \times \log(0)) \end{aligned}$$

$$\text{Entropy} = 0.9159$$

3. *Contrast*

Untuk menghitung nilai *Contrast*, digunakan Persamaan (2-8).

$$\begin{aligned} \text{Contrast} = & ((0 - 0)^2 \times 0.167) + ((0 - 1)^2 \times 0) + ((0 - 2)^2 \times 0.08) + \\ & ((0 - 3)^2 \times 0) + ((1 - 0)^2 \times 0) + ((1 - 1)^2 \times 0.167) + ((1 - 2)^2 \times \\ & 0.08) + ((1 - 3)^2 \times 0) + ((2 - 0)^2 \times 0.08) + ((2 - 1)^2 \times 0.08) + ((2 - \\ & 2)^2 \times 0.167) + ((2 - 3)^2 \times 0.08) + ((3 - 0)^2 \times 0) + ((3 - 1)^2 \times 0) + \\ & ((3 - 2)^2 \times 0.08) + ((3 - 3)^2 \times 0) \end{aligned}$$

$$\text{Contrast} = 0.96$$

4. *Homogeneity*

Untuk menghitung nilai *Homogeneity*, digunakan Persamaan (2-9).

$$\begin{aligned} \text{Homogeneity} = & \left(\frac{0.167}{1+|0-0|}\right) + \left(\frac{0}{1+|0-1|}\right) + \left(\frac{0.08}{1+|0-2|}\right) + \left(\frac{0}{1+|0-3|}\right) + \left(\frac{0}{1+|1-0|}\right) + \\ & \left(\frac{0.167}{1+|1-1|}\right) + \left(\frac{0.08}{1+|1-2|}\right) + \left(\frac{0}{1+|1-3|}\right) + \left(\frac{0.08}{1+|2-0|}\right) + \left(\frac{0.08}{1+|2-1|}\right) + \left(\frac{0.167}{1+|2-2|}\right) + \\ & \left(\frac{0.08}{1+|2-3|}\right) + \left(\frac{0}{1+|3-0|}\right) + \left(\frac{0}{1+|3-1|}\right) + \left(\frac{0.08}{1+|3-2|}\right) + \left(\frac{0}{1+|3-3|}\right) \end{aligned}$$

$$\text{Homogeneity} = 0.274$$

3.3.3 Klasifikasi

Proses klasifikasi yang dilakukan dalam penelitian ini menggunakan metode *k-nearest neighbor* (K-NN) dengan menggunakan nilai *k* sama dengan 3, 5 dan 7. Berikut ini urutan langkah dalam proses klasifikasi citra.

1. Misalkan terdapat nilai delapan fitur yaitu empat nilai fitur bentuk dan 4 nilai fitur tekstur, yang berdasarkan 6 data citra latih untuk 2 spesies daun yang masing-masing memiliki 8 ciri fitur dan pada sudut 90^0 . Dilihat pada Tabel 3.1.

Tabel 3.1 Nilai 4 fitur pada citra 2 spesies

Spesies	Canny				GLCM			
	<i>Area</i>	<i>Perimeter</i>	<i>Eccentricity</i>	<i>Matric</i>	<i>Energy</i>	<i>Entropy</i>	<i>Contrast</i>	<i>Homogeneity</i>
1	2129	1217	0.9586	0.0281	0.6962	1.822	0.7305	0.987
1	2960	1210	0.9578	0.0254	0.6984	1.818	0.7602	0.986
1	2266	1209	0.9507	0.0281	0.6941	1.819	0.7387	0.986
2	2801	1460	0.9303	0.0106	0.5641	2.8744	1.5625	0.977
2	2993	1461	0.9134	0.0141	0.5655	2.8645	1.5525	0.973
2	2902	1434	0.9101	0.0155	0.5585	2.8744	1.5622	0.970

2. Kemudian diketahui data uji pada satu citra memiliki nilai fitur bentuk *Area* =1925 , *Perimeter*=1435, *Eccentricity*=0.9450, *Matric*=0.182 dan fitur GLCM yaitu *Energy*=0.5677, *Entropy*=1.911, *Contrast* =0.922 dan *Homogeneity*=0.975.
3. Tentukan nilai parameter *k*, misalkan *k* sama dengan 3 dan kemudian dihitung jarak antara data *training* dengan data uji dengan menggunakan *cosine similarity* pada persamaan (2-10). Dapat dilihat pada Tabel 3.2.

Tabel 3.2 Menentukan nilai *Cosine similarity*

<i>Cosine similarity</i>	
$\frac{((2129 \times 1925) + (1217 \times 1435) + (0.9586 \times 0.9450) + (0.0281 \times 0.0182) + (0.6962 \times 0.5677) + (1.822 \times 1.911) + (0.7305 \times 0.922) + (0.987 \times 0.975))}{\sqrt{(2129^2 + 1217^2 + 0.9586^2 + 0.0281^2 + 0.6962^2 + 1.822^2 + 0.7305^2 + 0.987^2)} \times \sqrt{(1925^2 + 1435^2 + 0.9450^2 + 0.0182^2 + 0.5677^2 + 1.911^2 + 0.922^2 + 0.975^2)}}$	= 0.0020
$\frac{(2960 \times 1925) + (1210 \times 1435) + (0.9578 \times 0.9450) + (0.0254 \times 0.0182) + (0.6984 \times 0.5677) + (1.818 \times 1.911) + (0.7602 \times 0.922) + (0.986 \times 0.975)}{\sqrt{(2960^2 + 1210^2 + 0.9578^2 + 0.0254^2 + 0.6984^2 + 1.818^2 + 0.7602^2 + 0.986^2)} \times \sqrt{(1925^2 + 1435^2 + 0.9450^2 + 0.0182^2 + 0.5677^2 + 1.911^2 + 0.922^2 + 0.975^2)}}$	= 0.0026
$\frac{(2266 \times 1925) + (1209 \times 1435) + (0.9507 \times 0.9450) + (0.0281 \times 0.0182) + (0.6941 \times 0.5677) + (1.819 \times 1.911) + (0.920 \times 0.922) + (0.795 \times 0.975)}{\sqrt{(2266^2 + 1209^2 + 0.9507^2 + 0.0281^2 + 0.6941^2 + 1.819^2 + 0.7387^2 + 0.986^2)} \times \sqrt{(1925^2 + 1435^2 + 0.9450^2 + 0.0182^2 + 0.5677^2 + 1.911^2 + 0.922^2 + 0.975^2)}}$	= 0.00216
$\frac{(2801 \times 1925) + (1460 \times 1435) + (0.9303 \times 0.9450) + (0.0106 \times 0.0182) + (0.5641 \times 0.5677) + (2.8744 \times 1.911) + (1.5625 \times 0.922) + (0.977 \times 0.975)}{\sqrt{(2801^2 + 1460^2 + 0.9303^2 + 0.0106^2 + 0.5641^2 + 2.8744^2 + 1.5625^2 + 0.977^2)} \times \sqrt{(1925^2 + 1435^2 + 0.9450^2 + 0.0182^2 + 0.5677^2 + 1.911^2 + 0.922^2 + 0.975^2)}}$	= 0.00182
$\frac{(2993 \times 1925) + (1461 \times 1435) + (0.9134 \times 0.9450) + (0.0141 \times 0.0182) + (0.5655 \times 0.5677) + (2.8645 \times 1.911) + (1.5525 \times 0.922) + (0.973 \times 0.975)}{\sqrt{(2993^2 + 1461^2 + 0.9134^2 + 0.0141^2 + 0.5655^2 + 2.8645^2 + 1.5525^2 + 0.973^2)} \times \sqrt{(1925^2 + 1435^2 + 0.9450^2 + 0.0182^2 + 0.5677^2 + 1.911^2 + 0.922^2 + 0.975^2)}}$	= 0.00188
$\frac{(2902 \times 1925) + (1434 \times 1435) + (0.9101 \times 0.9450) + (0.0155 \times 0.0182) + (0.5585 \times 0.5677) + (2.8744 \times 1.911) + (1.5622 \times 0.922) + (0.970 \times 0.975)}{\sqrt{(2902^2 + 1434^2 + 0.9101^2 + 0.0155^2 + 0.5585^2 + 2.8744^2 + 1.5622^2 + 0.970^2)} \times \sqrt{(1925^2 + 1435^2 + 0.9450^2 + 0.0182^2 + 0.5677^2 + 1.911^2 + 0.922^2 + 0.975^2)}}$	= 0.00192

4. Setelah mengetahui *Cosine similarity* kemudian diurutkan untuk menentukan tetangga terdekat berdasarkan jarak maksimum mendekati satu k. Perhatikan Tabel 3.3.

Tabel 3.3 Menentukan yang termasuk 3-NN

<i>Cosine similarity</i>	Urutan jarak	Apakah termasuk 3-NN
0.0020	3	Iya($k=3$)
0.0026	1	Iya($k<3$)
0.00216	2	Iya($k<3$)
0.00182	6	Tidak($k>3$)
0.00188	5	Tidak($k>3$)
0.00192	4	Tidak($k>3$)

5. Menentukan kategori dari tetangga terdekat, perhatikan baris ke 1, 2, dan 3 pada Tabel 3.4. Kategori “Iya” diambil jika nilai $k \leq 3$. Jadi baris 1, 2, dan 3 termasuk tetangga terdekat. Perhatikan Tabel 3.4.

Tabel 3.4 Menentukan kategori spesies yang termasuk 3-NN

<i>Cosine similarity</i>	Urutan jarak	Apakah termasuk 3-NN	Spesies
0.0020	3	Iya($k=3$)	1
0.0026	1	Iya($k<3$)	1
0.00216	2	Iya($k<3$)	1
0.00182	6	Tidak($k>3$)	
0.00188	5	Tidak($k>3$)	
0.00192	4	Tidak($k>3$)	

6. Melihat dari kategori dominan spesies yang termasuk ke tetangga terdekat adalah spesies nomor 1 maka data citra uji disimpulkan adalah termasuk dalam spesies nomor 1.

3.3.4 Skenario Pengujian

Skenario pengujian dilakukan dengan dua skenario yang berbeda yakni :

1. Skenario pengujian pertama : Pada proses pelatihan, data yang digunakan yaitu dengan perbandingan 70:30, 70:20 dan 70:10 dengan data dipilih secara random.
2. Skenario pengujian kedua : Proses pengujian menggunakan *K-Fold Cross Validation*, digunakan *5-fold cross validation* dengan pengujian dilakukan sebanyak 5 kali. Variasi *cross validation* yang digunakan yaitu variasi *repeated* dengan indeks yang berbeda-beda dengan membagi *dataset* yang

berjumlah 350 data yang terbagi menjadi data *training* dan data *testing* menjadi 5 subset yang masing-masing subset berjumlah 70 data. Perhatikan Tabel 3.5.

Tabel 3.5 Skenario Uji Akurasi dengan 5 - *fold cross validation 1*

	1	2	3	4	5
5 Fold 1	<i>Test</i>	<i>Train</i>	<i>Train</i>	<i>Train</i>	<i>Train</i>
	<i>Train</i>	<i>Test</i>	<i>Train</i>	<i>Train</i>	<i>Train</i>
	<i>Train</i>	<i>Train</i>	<i>Test</i>	<i>Train</i>	<i>Train</i>
	<i>Train</i>	<i>Train</i>	<i>Train</i>	<i>Test</i>	<i>Train</i>
	<i>Train</i>	<i>Train</i>	<i>Train</i>	<i>Train</i>	<i>Test</i>

Tabel 3.6 Skenario Uji Akurasi dengan 5 - *fold cross validation 2*

	1	3	5	2	4
5 Fold 2	<i>Test</i>	<i>Train</i>	<i>Train</i>	<i>Train</i>	<i>Train</i>
	<i>Train</i>	<i>Test</i>	<i>Train</i>	<i>Train</i>	<i>Train</i>
	<i>Train</i>	<i>Train</i>	<i>Test</i>	<i>Train</i>	<i>Train</i>
	<i>Train</i>	<i>Train</i>	<i>Train</i>	<i>Test</i>	<i>Train</i>
	<i>Train</i>	<i>Train</i>	<i>Train</i>	<i>Train</i>	<i>Test</i>

Setelah menentukan jumlah citra latih dan citra uji, selanjutnya dilakukan *preprocessing* pada citra latih dan citra uji, ekstraksi fitur bentuk dengan *canny* dan ekstraksi fitur tekstur yang dilakukan pada sudut GLCM 90⁰ dan dengan menggunakan nilai k pada K-NN sama dengan 3, 5 dan 7 kemudian dari data tersebut akan dicari nilai akurasi yang paling optimal dari setiap citra yang diuji, dengan perhitungan akurasi yang dihitung dari pembagian seluruh data citra benar dibagi dengan keseluruhan data kemudian dikali 100%. Berikut ini perhitungan akurasi dimisalkan dari 10 data yang diuji ada 9 data yang benar [2].

$$\text{akurasi} = \frac{\text{jumlah data yang benar}}{\text{jumlah data keseluruhan}} \times 100\% \rightarrow \text{akurasi} = \frac{9}{10} \times 100\% = 90\%$$

3. Waktu Komputasi

Waktu komputasi merupakan metode yang dipilih untuk mengukur waktu yang dibutuhkan oleh model untuk melakukan proses pengenalan. Waktu komputasi ini meliputi waktu yang dibutuhkan oleh program untuk melakukan pelatihan dan juga pengujian. Waktu komputasi dihitung sejak proses pelatihan dan juga melakukan pengujian dimulai hingga proses terhenti. Dengan demikian dapat dihitung berapa waktu komputasi dari program yang dibangun.

DAFTAR PUSTAKA

- [1] F. Liantoni, “Klasifikasi Daun Dengan Perbaikan Fitur Citra Menggunakan Metode K-Nearest Neighbor,” *Ultim. Comput.*, 2015.
- [2] A. Hidayat and Makhsun, “Analisis Citra Daun Berdasarkan Fitur Local Binary Pattern Dan Fitur Canny Edge Detection Menggunakan Metode Klasifikasi K-Nearest Neighbor (K-NN),” *J. Esit*, pp. 1–10, 2018.
- [3] K. C. Daun, “Fitur berbasis fraktal dari koefisien wavelet untuk klasifikasi citra daun,” vol. 15, pp. 238–247, 2017.
- [4] M. Muchtar and L. Cahyani, “Klasifikasi Citra Daun dengan Metode Gabor Co-Occurrence,” *J. Ultim. Comput.*, vol. 7, no. 2, pp. 39–47, 2016.
- [5] H. Surahman, A. Fuja, and I. Rubandi, “Pengenalan Jenis Pempek Menggunakan Metode Canny & K-Nearest Neighbor (KNN) Berdasarkan Bentuknya,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 1, no. 14, pp. 1–11, 2017.
- [6] J. W. Yodha and A. W. Kurniawan, “Pengenalan Motif Batik Menggunakan Deteksi Tepi Canny Dan K-Nearest Neighbor,” *J. Teknol. Inf.*, vol. 13, no. 4, pp. 251–262, 2014.
- [7] R. K. Dewi and N. Suciati, “Klasifikasi Tanaman berdasarkan Fitur Bentuk dan Tekstur pada Daun menggunakan Decision Tree,” *J. Cybermatika*, vol. 3, no. 2, pp. 9–15, 2015.
- [8] F. Fatimah, G. F. Laxmi, and P. Eosina, “Pengubahan Data Image Ikan Air Tawar ke Data Vektor menggunakan Edge Detection Metode Canny,” *J. Ris. Pendidik. Mat.*, vol. 9, pp. 55–60, 2017.
- [9] P. Soepomo, “Implementasi Metode Canny Untuk Deteksi Tepi Mutu Daun Tembakau,” *J. Sarj. Tek. Inform.*, vol. 2, no. 3, pp. 231–243, 2014.
- [10] I. D. Kurniawati and A. Kusumawardhani, “Implementasi Algoritma Canny dalam Pengenalan Wajah menggunakan Antarmuka GUI Matlab,” no. December, pp. 3–8, 2017.

- [11] S. A. Wibowo, B. Hidayat, and U. Sunarya, "Simulasi dan Analisis Pengenalan Citra Daging Sapi dan Daging Babi dengan Metode GLCM," *J. Seniati*, no. 2, pp. 338–343, 2016.
- [12] M. Widyaningsih, "Identifikasi Kematangan Buah Apel Dengan Gray Level Co - Occurrence Matrix (GLCM)," *J. SAINTEKOM*, vol. 6, no. 1, pp. 71–88, 2016.
- [13] J. Kusanti and N. A. Haris, "Klasifikasi Penyakit Daun Padi Berdasarkan Hasil Ekstraksi Fitur GLCM Interval 4 Sudut," vol. 03, no. 01, pp. 1–6, 2018.
- [14] Subairi, Rahmadwati, and E. Yudaningtyas, "Implementasi Metode k-Nearest Neighbor pada Pengenalan Pola Tekstur Citra Saliva untuk Deteksi Ovulasi," *J. EECCIS*, vol. 12, no. 1, pp. 9–14, 2018.
- [15] D. A. N. Effendi and E. Z. Astuti, "Ekstraksi Fitur Histogram Guna Klasifikasi Jenis Citra Kayu Menggunakan Algoritma K-Nearest Neighbor," *J. Nas.*, pp. 1–13, 2017.
- [16] R. Andrian, S. Anwar, M. A. Muhammad, and A. Junaidi, "Identifikasi Kupu-Kupu Menggunakan Ekstraksi Fitur Deteksi Tepi (Edge Detection) dan Klasifikasi K-Nearest Neighbor (KNN)," *Jutisi (Jurnal Tek. Inform. dan Sist.*, vol. 5, pp. 234–243, 2019.
- [17] F. Flaurensia, T. Rismawan, and R. Hidayati, "Pengenalan Motif Batik Indonesia Menggunakan Deteksi Tepi Canny Dan Template Matching," *J. Coding, Sist. Komput. Untan*, vol. 4, no. 2, pp. 130–140, 2016.
- [18] I. Hastuti, "Perbandingan Metode Deteksi Tepi Menggunakan Metode Canny, Prewitt Dan Sobel Pada Image Ikan," *J. Repos. Poliban*, vol. 5662, pp. 9–10, 2016.
- [19] J. W. Yodha and A. W. Kurniawan, "Perbandingan Penggunaan Deteksi Tepi Dengan Metode Laplace, Sobel Dan Prewit Dan Canny Pada Pengenalan," *J. Teknol. Inf.*, vol. 13, no. 3, pp. 189–197, 2014.
- [20] L. H. Harefa, "Analisis Edge Detection Citra Digital Dengan Menggunakan

- Metode Robert Dan Canny,” *J. Ris. Komput.*, vol. 3, no. 1, pp. 29–34, 2016.
- [21] A. N. Hermana and M. S. Juerman, “Implementasi Algoritma Canny dan Backpropagation dalam Pengenalan Pola Rumah Adat,” *J. Itenas*, pp. 1–10, 2014.
- [22] N. C. Siantar, J. Hendryli, and D. E. Herwindiati, “Aplikasi Mobile Berbasis CBIR Untuk Pencarian Produk Ponsel dan Onlineshop,” *J. Ilmu Komput. dan Sist. Inf.*, vol. 7, no. 1, pp. 92–98, 2019.