IMPLEMENTASI ALGORITMA SUPPORT VECTOR MACHINE PADA KLASIFIKASI JENIS TANAMAN RIMPANG BERDASARKAN FITUR WARNA, BENTUK, DAN TEKSTUR

SKRIPSI

OLEH:
SANIA KAMELIA HAMDANI
NIM 19610071

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023

IMPLEMENTASI ALGORITMA SUPPORT VECTOR MACHINE PADA KLASIFIKASI JENIS TANAMAN RIMPANG BERDASARKAN FITUR WARNA, BENTUK, DAN TEKSTUR

SKRIPSI

Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan
dalam Memperoleh Gelar Sarjana Matematika (S.Mat)

Oleh
Sania Kamelia Hamdani
NIM. 19610071

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2023

IMPLEMENTASI ALGORITMA SUPPORT VECTOR MACHINE PADA KLASIFIKASI JENIS TANAMAN RIMPANG BERDASARKAN FITUR WARNA, BENTUK, DAN TEKSTUR

SKRIPSI

Oleh
Sania Kamelia Hamdani
NIM. 19610071

Telah Disetujui Untuk Diuji
Malang, 18 Desember 2023

Dosen Pembimbing I	Dosen Pembimbing II
Hisyam Fahmi, M.Kom.	M. Nafie Jauhari, M.Si.
NIP. 19890727 201903 1 018	NIP. 19870218 202321 1 018

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005

IMPLEMENTASI ALGORITMA SUPPORT VECTOR MACHINE PADA KLASIFIKASI JENIS TANAMAN RIMPANG BERDASARKAN FITUR WARNA, BENTUK, DAN TEKSTUR

SKRIPSI

Oleh
Sania Kamelia Hamdani
NIM. 19610071

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)

Tanggal 27 Desember 2023

Ketua Penguji	: Ari Kusumastuti, M.Si.	.....
Anggota Penguji 1	: Muhammad Khudzaifah, M.Si.	.....
Anggota Penguji 2	: Hisyam Fahmi, M.Kom.	.....
Anggota Penguji 3	: M. Nafie Jauhari, M.Si.	.....

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc
NIP. 19741129 200012 2 005

HALAMAN PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: Sania Kamelia Hamdani
NIM	: 19610071
Program Studi	: Matematika
Fakultas	: Sains dan Teknologi
Judul Skripsi	: Implementasi Algoritma *Support Vector Machine* pada Klasifikasi Jenis Tanaman Rimpang berdasarkan Fitur Warna, Bentuk, dan Tekstur

Menyatakan dengan sebenar-benarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan dan pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 27 Desember 2023
Yang membuat pernyataan,

Sania Kamelia Hamdani
NIM. 19610071

HALAMAN MOTO

“Work hard, dream big, never give up”

HALAMAN PERSEMBAHAN

Bismillahirrahmanirrahim, dengan mengucap syukur kepada Allah SWT skripsi ini peneliti persembahkan kepada ibunda Maryani, ayahanda Uuj Hamdanie, kakak dan adik yang senantiasa mendoakan dan memberi motivasi, nasihat, dan dukungan baik secara materiil maupun moril sehingga peneliti dapat menyelesaikan skripsi ini.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah, segala puji dan syukur senantiasa peneliti panjatkan kepada Allah *subhanahu wa ta'ala* atas segala rahmat, taufik, dan hidayah yang telah diberikan sehingga penulis dapat menyelesaikan skripsi dengan judul "Implementasi Algoritma *Support Vector Machine* pada Klasifikasi Janis Tanaman Rimpang berdasarkan Fitur Warna, Bentuk, dan Tekstur". Shalawat dan salam senantiasa tercurahkan kepada Nabi Muhammad *shallallahu 'alaihi wa sallam* yang telah membimbing manusia dalam jalan kebenaran, yakni agama Islam dan dinantikan syafaat beliau kelak di akhirat.

Dalam proses penyelesaian proposal skripsi ini, peneliti banyak mendapat bimbingan, dukungan, dan arahan serta sumbangsih dari berbagai pihak. Oleh karena itu peneliti mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. M. Zainuddin, M.A., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si., selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Elly Susanti, M.Sc., selaku Ketua Program Studi Matematika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Hisyam Fahmi, M.Kom., selaku dosen pembimbing I yang telah memberikan berbagai pengetahuan, pengalaman, arahan, nasihat, serta motivasi kepada peneliti.
5. M. Nafie Jauhari, M.Si., selaku dosen pembimbing II yang telah memberikan ilmu, nasihat, bimbingan, pengalaman, serta motivasi kepada peneliti.
6. Seluruh dosen Program Studi Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
7. Kedua orang tua serta seluruh keluarga yang senantiasa mendoakan, memberikan semangat, dukungan, nasihat, serta kasih sayang sehingga peneliti dapat menyelesaikan tugas akhir.

8. Rekan peneliti yang berjuang bersama dalam mengerjakan skripsi yang senantiasa memberikan dukungan dan semangat kepada peneliti.
9. Seluruh teman-teman Program Studi Matematika angkatan 2019 yang senantiasa memberikan bantuan, dukungan dan semangat kepada peneliti dalam berbagai kondisi.

Semoga Allah *subhanahu wa ta'ala* selalu memberikan balasan atas segala bantuan dan kebaikan yang telah diberikan kepada penulis. Penulis berharap agar laporan ini dapat bermanfaat bagi penulis serta pembaca untuk menambah wawasan keilmuan yang selalu berkembang.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 27 Desember 2023

Penulis

DAFTAR ISI

HALAMAN JUDUL ..... **i**
HALAMAN PENGAJUAN ..... **ii**
HALAMAN PERSETUJUAN ..... **iii**
HALAMAN PENGESAHAN ..... **iv**
HALAMAN PERNYATAAN KEASLIAN TULISAN ..... **v**
HALAMAN MOTO ..... **vi**
HALAMAN PERSEMBAHAN ..... **vii**
KATA PENGANTAR ..... **viii**
DAFTAR ISI ..... **x**
DAFTAR GAMBAR ..... **xii**
DAFTAR TABEL ..... **xiii**
DAFTAR LAMPIRAN ..... **xiv**
ABSTRAK ..... **xv**
ABSTRACT ..... **xvi**
مستخلص البحث ..... **xvii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Tujuan Penelitian 3
1.4 Manfaat Penelitian 3
1.5 Batasan Masalah 4
1.4 Definisi Istilah 4
BAB II KAJIAN PUSTAKA **6**
2.1 Teori Pendukung 6
2.1.1 *Machine Learning* 6
2.1.2 *Support Vector Machine* 7
2.1.3 Evaluasi 11
2.1.4 Citra Digital 13
2.1.5 Segmentasi Citra 14
2.1.6 Ekstraksi Fitur Warna 18
2.1.7 Ekstraksi Fitur Bentuk 18
2.1.8 Ekstraksi Fitur Tekstur 20
2.1.9 Rimpang 23
2.2 Kajian Integrasi Topik dengan Al-Qur'an/Hadits 25
2.3 Kajian Topik dengan Teori Pendukung 27
BAB III METODE PENELITIAN **30**
3.1 Jenis Penelitian 30
3.2 Data dan Sumber Data 30
3.3 Tahapan Penelitian 31
BAB IV METODE PENELITIAN **35**
4.1 Segmentasi Citra 35
4.2 Ekstraksi Fitur 39
4.2.1 Ekstraksi Fitur Warna 40
4.2.2 Ekstraksi Fitur Bentuk 42

4.2.3 Ekstraksi Fitur Tekstur 44
4.3 Klasifikasi Citra 47
4.4 Evaluasi 52
4.5 Implementasi *Support Vector Machine* dalam Pandangan Islam 54
BAB V KESIMPULAN DAN SARAN 57
5.1 Kesimpulan 57
5.2 Saran 57
DAFTAR PUSTAKA 59
LAMPIRAN 62
RIWAYAT HIDUP 68

DAFTAR GAMBAR

Gambar 2.1	*Hyperplane* Pada Algoritma SVM	7
Gambar 2.2	Contoh Data Yang Tidak Bisa Dipisahkan Linear	8
Gambar 2.3	Contoh Penerapan *Soft Margin* SVM	9
Gambar 2.4	SVM Pada Kasus Lima Kelas	11
Gambar 2.5	Pembagian Data Metode *K-fold Cross Validation*	13
Gambar 2.6	Matriks Representasi Citra Digital	13
Gambar 2.7	*Eccentricity* Dari Sebuah Elips	19
Gambar 2.8	Sudut Yang Dapat Digunakan Pada GLCM	20
Gambar 2.9	Proses Perhitungan Jumlah Hubungan Dua Piksel	21
Gambar 2.10	Jenis-Jenis Tanaman Rimpang	24
Gambar 3.1	Diagram Alir Penelitian	33
Gambar 4.1	Citra Sampel	35
Gambar 4.2	Citra Rimpang Dalam Format HSV	37
Gambar 4.3	Hasil Segmentasi K-means	38
Gambar 4.4	Citra Hasil *Thresholding*	38
Gambar 4.5	Hasil Segmentasi Citra Rimpang	39
Gambar 4.6	Hasil Pemisahan Fitur RGB	41
Gambar 4.7	Citra Sampel Biner	42
Gambar 4.8	Visualisasi *Eccentricity*	43
Gambar 4.9	Contoh Pembagian Kelas Berdasarkan Hasil Klasifikasi	50

DAFTAR TABEL

Tabel 4.1 Tabel Hasil Proses Klasifikasi Jenis Tanaman Rimpang..... 51

DAFTAR LAMPIRAN

Lampiran 1	Matriks Citra Sampel	62
Lampiran 2	Matriks Citra Sampel berformat *Grayscale*	62
Lampiran 3	Matriks GLCM Non Simetris.....	62
Lampiran 4	Matriks GLCM Simetris	62
Lampiran 5	Matriks GLCM Ternormalisasi.....	62
Lampiran 6	Hasil Klasifikasi	62
Lampiran 7	Kode *Python*.....	62

ABSTRAK

Hamdani, Sania Kamelia. 2023. **Implementasi Algoritma *Support Vector Machine* pada Klasifikasi Jenis Tanaman Rimpang berdasarkan Fitur Warna, Bentuk, dan Tekstur**. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Hisyam Fahmi, M.Kom. (II) M. Nafie Jauhari, M.Si.

Kata Kunci: *Machine Learning*, Klasifikasi, *Image Processing*, *Support Vector Machine*, Tanaman Rimpang

Klasifikasi merupakan sebuah proses pembelajaran *machine learning* yang memiliki tujuan untuk mengkategorikan data berdasarkan kelasnya. Salah satu metode klasifikasi yang dapat digunakan yaitu *support vector machine* (SVM). Pada penelitian ini, metode SVM digunakan untuk mengklasifikasikan jenis tanaman rimpang. Tanaman rimpang merupakan jenis tanaman yang merambat di bawah permukaan tanah. Pada umumnya, tanaman rimpang memiliki warna, bentuk, dan tekstur yang cenderung sama. Oleh karena itu, penelitian ini memiliki tujuan untuk menghasilkan model klasifikasi yang dapat membantu manusia dalam membedakan jenis tanaman dengan mempertimbangkan hasil tingkat akurasinya. Adapun jenis tanaman rimpang yang digunakan pada penelitian ini terdapat 5 jenis yaitu jahe, lengkuas, temulawak, kencur, dan kunyit. Data yang digunakan dalam penelitian ini diambil dari *website* Kaggle dimana terdapat 125 data citra. Penelitian ini secara umum terbagi menjadi empat bagian yaitu *preprocessing*, ekstraksi fitur, klasifikasi menggunakan SVM serta evaluasi dengan algoritma K-*fold Cross Validation*. Tahap *preprocessing* pada penelitian ini terdiri dari segmentasi citra yang bertujuan untuk memisahkan objek rimpang dengan objek lainnya pada citra. Sedangkan tahap ekstraksi fitur dilakukan secara bertahap untuk dapat mendapatkan fitur warna, bentuk, dan tekstur. Proses klasifikasi dilakukan sehingga menghasilkan 10 model SVM. Adapun pada proses evaluasi, digunakan nilai $k = 5$. Hasil dari penelitian ini diperoleh tingkat akurasi dari implementasi algoritma *support vector machine* pada klasifikasi jenis tanaman rimpang berdasarkan fitur warna, bentuk, dan tekstur ialah sebesar 96,8%.

ABSTRACT

Hamdani, Sania Kamelia. 2023. **The Implementation of Support Vector Machine Algorithm on Classification of Rhizome Types based on Color, Shape, and Texture Features**. Undergraduate Thesis. Mathematics Department, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) Hisyam Fahmi, M.Kom. (II) M. Nafie Jauhari, M.Si.

Keywords: Machine Learning, Classification, Image Processing, Support Vector Machine, Rizhome

Classification is a machine learning process that aims to categorize data based on its class. One of the classification methods that can be used is support vector machine (SVM). In this research, the SVM method is used to classify the types of rhizome plants. Rhizome plants are a type of plant that creeps below ground level. In general, rhizome plants have the same color, shape, and texture. Therefore, this research aims to produce a classification model that can help humans in distinguishing plant species by considering the results of its accuracy level. There are five types of rhizome plants used in this study, namely ginger, galangal, curcuma, aromatic ginger, and turmeric. The data used in this research is taken from the Kaggle website where there are 125 image data. This research is generally divided into four parts, namely preprocessing, feature extraction, classification using SVM and evaluation with the K-fold Cross Validation algorithm. The preprocessing stage in this study consists of image segmentation which aims to separate rhizome objects from other objects in the image. Meanwhile, the feature extraction stage is carried out in stages to obtain color, shape, and texture features. The classification process is carried out to produce 10 SVM models. As for the evaluation process, the value of $k = 5$. The results of this study obtained the accuracy rate of the implementation of the support vector machine algorithm on the classification of rhizome plant species based on color, shape, and texture features is 96.8%.

مستخلص البحث

هدني، سانيا كميليا. ٢٠٢٣. **تنفيذ خوارزمية الة المتجه الداعم في تصنيف أنواع النباتات الدرنية استنادًا إلى ميزات اللون والشكل والملمس**. البحث الجامعي. قسم الرياضيات،كلية العلوم والتكلنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (١) هشام فهمي، الماجستير، (٢) محمد نافع جوهري، الماجستير

الكلمات الرئيبة: التعلم الآلي، تصنيف، معالجة الصورة، الة المتجه الداعم، نباتات الجذمور

التصنيف هو عملية تعلم آلي تهدف إلى تصنيف البيانات بناء على فئتها. إحدى طرق التصنيف التي يمكن استخدامها هي آلة المتجه الدٰعم *(SVM)*. في هذا البحث، تم استخدام طريقة *SVM* لتصنيف أنواع نباتات الجذمور. نباتات جذمور هي نوع من النباتات التي تنتشر تحت سطح التربة. بشكل عام، نباتات جذمور لها نفس اللون والشكل والملمس. لذلك يهدف هذا البحث إلى إنتاج نموذج تصنيف يمكن أن يساعد الإنسان في تمييز أنواع النباتات من خلال النظر في نتائج مستوى دقتها. هناك ٥ أنواع من نباتات الجذور المستخدمة في هذا البحث، وهي الزنجبيل والخولنجان والكركم الجاف والزنجبيل العطري والكركم. تم أخذ البيانات المستخدمة في هذا البحث من موقع كاغل حيث كان هناك ١٢٥ بيانات صورة. ينقسم هذا البحث بشكل عام إلى أربعة أجزاء، وهي المعالجة المسبقة، واستخراج الميزات، والتصنيف باستخدام *SVM* والتقييم باستخدام خوارزمية التحقق المتقاطع ك-طيه. تتكون مرحلة المعالجة المسبقة في هذا البحث من تجزئة الصورة التي تهدف إلى فصل كائنات جذمور عن كائنات أخرى في الصورة. بينما يتم تنفيذ مرحلة استخراج الميزات تدريجيا لتتمكن من الحصول على ميزات اللون والشكل والملمس. تم تنفيذ عملية التصنيف مما أدى إلى ١٠ نماذج *SVM*. بالنسبة لعملية التقييم، يتم استخدام قيمة ك = ٥ بحيث يكون هناك ٥ تجارب. حصلت نتائج هذا البحث على معدل دقة تنفيذ خوارزمية آلة المتجه الدٰعم في تصنيف أنواع نباتات جذمور بناء على خصائص اللون والشكل والملمس بنسبة ٩٦.٨%.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi yang berkembang pesat saat ini dapat memberikan manfaat dan membantu masyarakat dalam kehidupan sehari-hari. Salah satu bidang teknologi yang dapat digunakan untuk membantu kehidupan sehari-hari adalah *machine learning*. *Machine learning* merupakan salah satu bidang teknologi yang dapat membantu sistem memiliki kecerdasan. *Machine learning* memiliki tujuan untuk mempelajari hubungan antara *input* dan *output* (Kusuma, 2020). *Machine learning* sering digunakan untuk menangani proses klasifikasi atau pengenalan pola. Klasifikasi merupakan proses pengelompokkan suatu data terhadap kategori tertentu, dengan tujuan untuk memprediksi data baru.

Salah satu metode klasifikasi pada *machine learning* adalah *Support vector machine* (SVM). Metode SVM termasuk dalam jenis *supervised learning*, yaitu jenis *machine learning* yang dilakukan dengan cara mempelajari hubungan antara data dengan labelnya. Tujuan algoritma SVM adalah untuk menemukan *hyperplane*, yaitu garis yang digunakan untuk memisah antar kelas. Algoritma SVM dapat mencari *hyperplane* yang optimal dengan mengoptimalkan jarak antara titik terluar dari data terhadap *hyperplane* tersebut. Adapun titik terluar dari data disebut sebagai *support vector*.

Sebagai negara agraris, Indonesia dikenal memiliki beragam jenis rempah-rempah seperti jahe, kunyit, kencur, temulawak, dan sebagainya. Hal ini tentu saja didukung oleh keunggulan geografis Indonesia. Oleh karena itu, Indonesia

mendapat julukan *"mother of spices"* (ibu rempah-rempah) dunia (Setiawan, 2020). Secara umum tanaman rimpang memiliki warna, bentuk, dan tekstur yang hampir mirip. Hal ini membuat sebagian besar orang mengalami kesulitan dalam membedakan jenis tanaman rimpang. Oleh karena itu, diperlukan sebuah teknologi kecerdasan buatan seperti *machine learning* yang dapat membantu manusia dalam membedakan jenis tanaman rimpang.

Dalam agama Islam, menuntut ilmu merupakan suatu kewajiban bagi setiap muslim. Mengembangkan ilmu merupakan salah satu cara bagi manusia untuk bersyukur atas segala ciptaan Allah SWT dan mengagungkan kebesaran Allah SWT. Hal ini bersesuaian dengan Al-Qur'an surat Ali Imran ayat 190 yang berbunyi (Kemenag RI, 2022):

اِنَّ فِيْ خَلْقِ السَّمٰوٰتِ وَالْاَرْضِ وَاخْتِلَافِ الَّيْلِ وَالنَّهَارِ لَاٰيٰتٍ لِّاُولِى الْاَلْبَابِۙ

"Sesungguhnya dalam penciptaan langit dan bumi serta pergantian malam dan siang terdapat tanda-tanda (kebesaran Allah) bagi orang yang berakal," (Q.S. Ali Imran:190)

Dalam ayat tersebut dijelaskan bahwa segala ciptaan Allah SWT baik di langit maupun di bumi merupakan bentuk kebesaran Allah SWT. Untuk itu, manusia diberikan akal oleh Allah SWT untuk mengkaji segala ciptaan Allah SWT. Dengan pengkajian atau penelitian tersebut, diharap manusia dapat menemukan tanda-tanda kebesaran Allah SWT sehingga meningkatkan iman dan taqwa terhadap Allah SWT. Dengan demikian, penelitian ini juga merupakan salah satu bentuk karya untuk mengagungkan kebesaran Allah SWT.

Pada penelitian sebelumnya, sudah diteliti proses klasifikasi jenis rimpang berdasarkan fitur warna dan tekstur menggunakan metode KNN. Metode KNN merupakan metode klasifikasi yang dilakukan berdasarkan jarak ketetanggaan dari

suatu data terhadap data lainnya. Pada penelitian tersebut, dihasilkan nilai akurasi 73% dengan menggunakan nilai ketetanggaan (k) yaitu 1 (Kaharuddin, dkk., 2019).

Pada penelitian ini, algoritma yang akan digunakan dalam proses klasifikasi adalah algoritma *Support Vector Machine* (SVM). Keunggulan SVM adalah memiliki akurasi yang cukup baik meskipun dilatih dengan data yang sedikit (Ichwan, dkk., 2018). Dengan menggunakan metode SVM diharapakan hasil penelitian ini dapat memiliki tingkat akurasi yang tinggi sehingga dapat membantu masyarakat dalam kehidupan sehari-hari.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan rumusan masalah yaitu berapa nilai tingkat akurasi klasifikasi jenis tanaman rimpang berdasarkan warna, bentuk, dan tekstur dengan menggunakan algoritma SVM?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas maka tujuan penelitian ini adalah untuk mengetahui nilai tingkat akurasi klasifikasi jenis tanaman rimpang berdasarkan warna, bentuk, dan tekstur dengan menggunakan algoritma SVM.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini berdasarkan uraian-uraian di atas yaitu untuk menambah wawasan terkait efektivitas algoritma SVM dalam klasifikasi jenis tanaman rimpang berdasarkan hasil akurasinya. Selain itu, hasil yang diperoleh dari

penelitian ini juga dapat digunakan untuk membantu masyarakat dalam membedakan jenis tanaman rimpang.

1.5 Batasan Masalah

Batasan masalah yang digunakan pada penelitian ini adalah sebagai berikut.

1. File citra tanaman rimpang hanya memiliki fitur RGB atau HSV.
2. Jenis tanaman rimpang yang dipilih sesuai dengan dataset yaitu jahe, kencur, temulawak, kunyit, dan lengkuas karena sumber data ini diambil dari kaggle.
3. Fitur yang digunakan yaitu fitur warna, bentuk, dan tekstur dikarenakan ketiga fitur tersebut merupakan fitur yang dapat digunakan dan diambil dari data berbentuk citra.

1.4 Definisi Istilah

Terdapat beberapa istilah yang digunakan pada penelitian ini sebagai berikut:

Data Latih	: Data yang digunakan untuk melatih algoritma program untuk mendapatkan model yang sesuai
Data Uji	: Data yang digunakan untuk menguji model dan mengetahui tingkat keakuratan
Learning Rate	: Salah satu parameter *training* untuk menghitung nilai koreksi bobot pada proses pelatihan
Preprocessing	: Proses pengolahan data untuk mengubah data mentah ke dalam bentuk yang lebih mudah dipahami sistem

Hyperplane	:	Garis yang digunakan untuk memisah antar kelas dalam ruang dimensi *M*
Support Vector	:	Titik terluar dari data
Akurasi	:	Nilai yang dihasilkan melalui perbandingan banyak klasifikasi yang bernilai benar dengan banyak seluruh data yang diklasifikasi
Rimpang	:	Batang tanaman yang merambat di bawah permukaan tanah dan dapat menghasilkan tunas dan akar baru dari ruas-ruasnya

BAB II

KAJIAN PUSTAKA

2.1 Teori Pendukung

2.1.1 *Machine Learning*

Machine learning merupakan salah satu bagian dari *artificial intelligence* (kecerdasan buatan) yang dapat digunakan untuk melatih suatu sistem atau model untuk mempelajari data tanpa menuliskan algoritma secara eksplisit (Kusuma, 2020). Dalam penerapannya *machine learning* sering digunakan dalam proses klasifikasi, yaitu pengelompokkan data berdasarkan kategori tertentu untuk selanjutnya digunakan untuk mendeteksi kategori dari data baru.

Secara umum, *machine learning* dibagi menjadi tiga jenis pendekatan yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning* (Murphy, 2012).

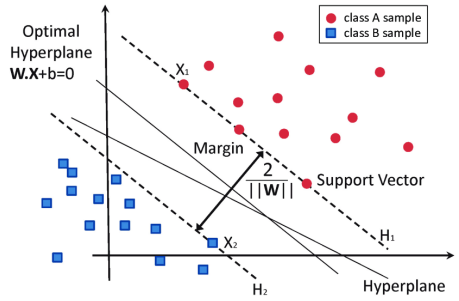
Supervised learning merupakan jenis *machine learning* yang dilakukan menggunakan data yang telah diberi label. Tujuan dari pendekatan ini adalah untuk mempelajari pola hubungan antara *input* x dengan *output* y dimana $y \in \{1, \ldots, C\}$ dengan C adalah banyak kelas atau kategori. Pada *supervised learning*, *output* y juga umum disebut sebagai label dari *input* x. *Supervised learning* pada umumnya sering digunakan untuk keperluan klasifikasi, pengenalan pola, atau regresi.

Pada *supervised learning*, diperlukan dua data yaitu data latih dan data uji. Data latih merupakan pasangan *input-output* $D = \{(x_i, y_i)\}_{i=1}^{N}$ dimana N merupakan banyak data latih. Data latih merupakan data yang digunakan untuk melatih sistem untuk mendapatkan model terbaik. Hasil dari proses latih ini

selanjutnya dapat digunakan untuk memprediksi *output* baik berupa kelas atau kategori dari data baru. Data baru yang digunakan untuk menguji model *supervised learning* disebut sebagai data uji.

Setiap data x_i yang digunakan dalam *supervised learning* merupakan sebuah vektor dengan dimensi *D* yang merepresentasikan suatu ciri dari data tersebut. Vektor ini juga sering disebut sebagai fitur. Pada umumnya, data x_i yang digunakan berasal dari objek terstruktur seperti citra (gambar), teks, grafik, dll.

2.1.2 *Support Vector Machine*



Gambar 2.1 *Hyperplane* Pada Algoritma SVM
Sumber: Garcia-Gonzalo, 2016

Support vector machine (SVM) adalah satu jenis *supervised learning* yang dapat digunakan untuk proses klasifikasi (Abe, 2005). Tujuan algoritma SVM adalah untuk menemukan *hyperplane*, yaitu garis yang digunakan untuk memisah antar kelas dalam ruang dimensi *M* seperti terlihat pada Gambar 2.1. *Support vector machine* dapat mencari *hyperplane* yang optimal dengan mengoptimalkan jarak antara titik terluar dari data (titik terdekat dengan *hyperplane*) terhadap *hyperplane*. Titik terluar dari data disebut sebagai *support vector*. Sedangkan, jarak antara *support vector* terhadap *hyperplane* disebut sebagai *margin*. Fungsi *hyperplane* pada umumnya ditulis dalam Persamaan (2.1).

$$D(x) = w^T x + b \tag{2.1}$$

dimana w merupakan vektor bobot, x merupakan vektor data dan b merupakan vektor bias. Suatu data x dapat diklasifikasikan ke dalam suatu kelas jika $D(x) \geq 1$ atau diklasifikasikan ke dalam kelas lainnya jika $D(x) \leq -1$. Dengan demikian, secara umum setiap data latih memenuhi Persamaan (2.2).

$$\frac{y_i(w^T x_i + b)}{\|w\|} \geq \delta, \qquad i = 1,2, \dots, M \tag{2.2}$$

dimana δ adalah *margin*, x_i adalah data ke-i dan y_i adalah label dari x_i.

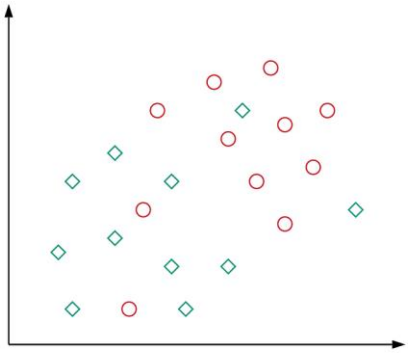
Hyperplane optimal secara matematis dapat ditemukan dengan meminimalkan hasil Persamaan (2.3).

$$Q(w) = \frac{1}{2}\|w\|^2 \tag{2.3}$$

dengan syarat tertulis pada Persamaan (2.4).

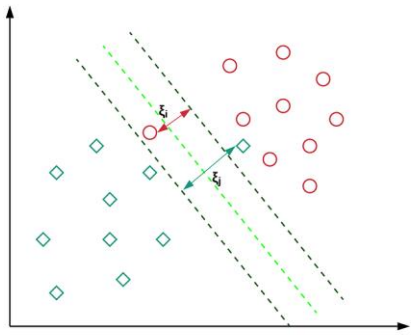
$$y_i(w^T x_i + b) \geq 1, \qquad i = 1,2, \dots, M \tag{2.4}$$

Pada kasus linear, seringkali terjadi *overfitting*, yaitu kondisi dimana model hanya bekerja dengan baik terhadap data latih namun memiliki banyak kesalahan jika diterapkan pada data uji (Misra, 2019). Hal ini dikarenakan tidak setiap data dapat dipisahkan dengan linear secara langsung. Contoh data yang tidak dapat dipisahkan secara langsung dapat dilihat pada Gambar 2.2.



Gambar 2.2 Contoh Data Yang Tidak Bisa Dipisahkan Secara Linier
Sumber: Misra, 2019

Pada data tersebut, terdapat beberapa titik data yang cenderung masuk ke dalam wilayah lain. Oleh karena itu, perlu dilakukan pengembangan model SVM. Pengembangan SVM yang sering digunakan dikenal sebagai *soft margin*. Penerapan *soft margin* dilakukan untuk menjaga jarak margin sehingga margin tetap lebar untuk mencegah *overfitting*. Hasil proses *soft margin* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Contoh Penerapan *Soft Margin* SVM
Sumber: Garcia-Gonzalo, 2016

Dengan menerapkan *soft margin*, maka *hyperplane* optimal dapat dicari dengan meminimalkan Persamaan (2.5).

$$Q(w) = \frac{1}{2}\|w\|^2 + \mathrm{C}\sum_i \xi_i \tag{2.5}$$

dimana ξ_i merupakan jarak titik x_i terhadap *hyperplane* kelasnya dan C merupakan konstanta. Persamaan ini digunakan dengan memperhatikan syarat tertulis pada Persamaan (2.6).

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1,2,\dots,M \tag{2.6}$$

Dengan memperhatikan Persamaan (2.6), dapat diperoleh Persamaan (2.7).

$$\xi_i = \begin{cases} 1 - y_i(w^T x_i + b) & , y_i(w^T x_i + b) < 1 \\ 0 & , y_i(w^T x_i + b) \geq 1 \end{cases} \tag{2.7}$$

Persamaan (2.7) juga dapat ditulis sebagai Persamaan (2.8).

$$\xi_i = max(1 - y_i(w^T x_i + b), 0) \quad (2.8)$$

Dengan demikian, kita dapat menulis ulang Persamaan (2.5) menjadi Persamaan (2.9).

$$Q(w, b) = \frac{1}{2}\|w\|^2 + \mathrm{C}\sum_i max(1 - y_i(w^T x_i + b), 0) \quad (2.9)$$

Selanjutnya, untuk mendapatkan solusi optimal dapat dilakukan dengan meminimalkan Persamaan (2.9) terhadap w dan b. Dengan demikian, diperoleh turunan parsial Persamaan (2.9) terhadap w dan b masing-masing pada Persamaan (2.10) dan (2.11).

$$\frac{\partial Q(w, b)}{\partial w} = \|w\| + \mathrm{C}\sum_i \frac{\partial[max(1 - y_i(w^T x_i + b), 0)]}{\partial w} \quad (2.10)$$

$$\frac{\partial Q(w, b)}{\partial b} = \mathrm{C}\sum_i \frac{\partial[max(1 - y_i(w^T x_i + b), 0)]}{\partial b} \quad (2.11)$$

Dengan menggunakan algoritma *gradient descent*, maka proses pencarian solusi optimal dapat dilakukan secara berulang menggunakan Persamaan (2.13) dan (2.14).

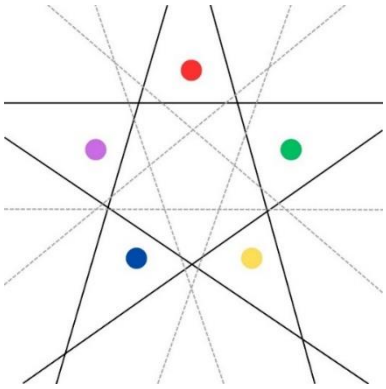
$$w_j = w_{j-1} + \lambda\left(\|w\| + \mathrm{C}\sum_i \frac{\partial[max(1 - y_i(w^T x_i + b), 0)]}{\partial w}\right) \quad (2.12)$$

$$b_j = b_{j-1} + \lambda\left(\mathrm{C}\sum_i \frac{\partial[max(1 - y_i(w^T x_i + b), 0)]}{\partial b}\right) \quad (2.13)$$

λ merupakan nilai *learning rate*, w_j merupakan nilai bobot ke-j yang diperoleh dari perhitungan Persamaan (2.12) menggunakan nilai bobot sebelumya w_{j-1}, dan b_j juga merupakan nilai bias ke-j yang diperoleh dari perhitungan Persamaan (2.13) dengan menggunakan nilai bias sebelumnya b_{j-1}.

Pada dasarnya, SVM hanya dapat digunakan untuk klasifikasi biner (dua kelas). Pada kasus klasifikasi sebanyak n kelas, SVM akan membuat sebanyak $\frac{n(n-1)}{2}$ model SVM. Pada penelitian ini, banyak kelas yang digunakan terdapat 5 kelas. Dengan demikian, banyak *hyperplane* yang diperlukan dalam proses

klasifikasi adalah sebanyak $\frac{5(5-1)}{2} = 10$ model SVM. Secara visual, model SVM tersebut dapat dilihat pada Gambar 2.4 dimana garis terhubung merupakan *hyperplane* pemisah antar kelas terhadap seluruh kelas lainnya dan garis terpisah merupakan *hyperplane* pemisah antar dua kelas terhadap tiga kelas lainnya.



Gambar 2.4 SVM Pada Kasus Lima Kelas

2.1.3 Evaluasi

Pada penelitian ini, proses evaluasi dilakukan untuk menilai efektivitas model *machine learning* yang digunakan dalam proses implementasi SVM dalam klasifikasi jenis tanaman rimpang berdasarkan fitur warna, bentuk, dan tekstur. Metrik yang digunakan untuk evaluasi dalam penelitian ini yaitu nilai akurasi. Nilai akurasi merupakan nilai yang dihasilkan melalui perbandingan banyak klasifikasi yang bernilai benar dengan banyak seluruh data yang diklasifikasi (Hasty, 2023). Secara matematis, akurasi dapat ditulis dalam Persamaan (2.14).

$$akurasi = \frac{Jumlah\ klasifikasi\ bernilai\ benar}{Jumlah\ total\ klasifikasi} \tag{2.14}$$

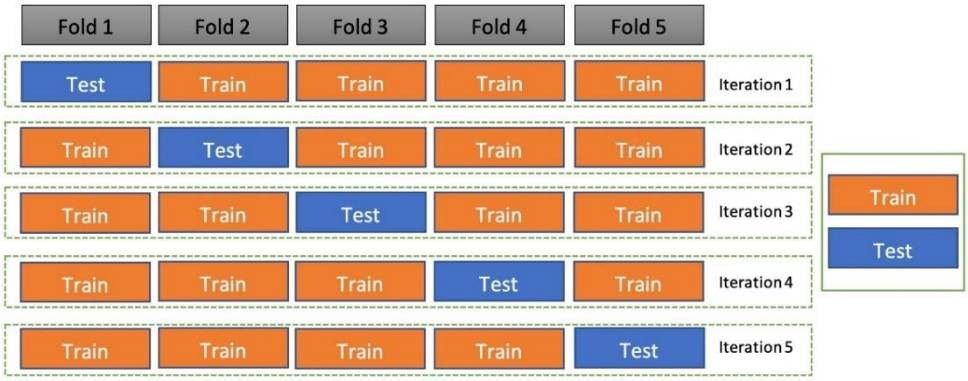
Pada penelitian ini, untuk mengetahui jumlah klasifikasi bernilai benar dapat dilakukan dengan menggunakan *confusion matrix*. *Confusion matrix* merupakan matriks persegi yang merepresentasikan hubungan nilai/label asli suatu

data terhadap hasil prediksi/klasifikasinya (Kundu, 2022). Pada umumnya, kolom pada *confusion matrix* merepresentasikan label asli suatu data, sedangkan baris merepresentasikan hasil prediksinya. Contoh *confusion matrix* dapat dilihat pada Gambar 2.5.

Gambar 2.5 *Confusion Matrix*
Sumber: Kundu, 2022

Pada Gambar 2.5, kita dapat melihat berapa banyak data yang terklasifikasikan dengan benar dan berapa banyak yang tidak terklasifikasikan dengan benar dengan jelas. Pada contoh tersebut, diperoleh jumlah klasifikasi bernilai benar adalah 191 data dan jumlah total klasifikasi adalah 145 data. Dengan menggunakan Persamaan (2.14) maka diperoleh nilai akurasi pada contoh tersebut yaitu 75,92%.

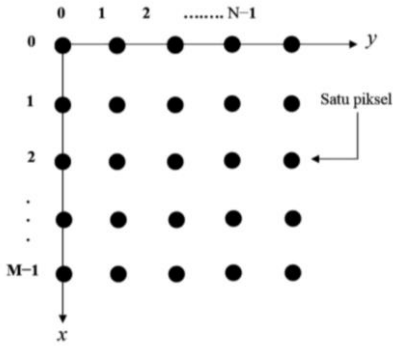
Secara keseluruhan, proses evaluasi akan dilakukan menggunakan metode *K-fold Cross Validation*, yaitu proses evaluasi dengan cara membagi data menjadi k bagian dengan banyak data pada tiap bagian sebesar $\frac{1}{k}$ dari jumlah seluruh data (Pandian, 2009). Visualisasi pembagian data pada metode *K-fold Cross Validation* dapat dilihat pada Gambar 2.6.



Gambar 2.6 Pembagian Data Metode *K-fold Cross Validation*
Sumber: Ranjan, 2021

Pada metode *K-fold Cross Validation*, data akan dilatih dan diuji sebanyak k kali dengan menggunakan $k - 1$ data sebagai data latih dan 1 bagian lainnya sebagai data uji dimana data latih dan data uji akan digilir terhadap masing-masing bagian. Dengan demikian, seluruh data memiliki peluang yang sama untuk menjadi data latih maupun data uji. Hal ini juga dapat menghasilkan nilai evaluasi yang lebih akurat.

2.1.4 Citra Digital



Gambar 2.7 Matriks Representasi Citra Digital
Sumber: Andono, 2017

Citra digital merupakan citra yang dapat direpresentasikan dan diolah oleh komputer (Andono, 2017). Citra digital memiliki bentuk seperti matriks berukuran

M × N seperti terlihat pada Gambar 2.7. Pada citra digital, setiap elemen matriks disebut sebagai piksel (*picture element*).

Setiap piksel pada citra merepresentasikan intensitas warna citra pada suatu titik dengan koordinat (x, y). Pada umumnya, nilai intensitas warna pada citra dapat berupa RGB, HSV, *grayscale*, atau juga biner. Pada citra RGB, setiap piksel pada citra memiliki tiga nilai yang masing-masing mewakili warna merah, hijau, dan biru. Pada citra *grayscale* setiap piksel pada citra mewakili intensitas cahaya dengan rentang 0 hingga 255. Sedangkan pada citra biner, setiap piksel pada citra hanya dapat diisi oleh dua angka saja (umumnya 0 dan 1). Semakin tinggi nilai intensitas pada piksel, maka semakin cerah piksel tersebut.

Dengan demikian, citra digital dapat direpresentasikan dalam matriks sebagai berikut.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f\big(0,(M-1)\big) \\ f(1,0) & f(1,1) & \cdots & f\big(1,(M-1)\big) \\ \vdots & \vdots & \ddots & \vdots \\ f\big((N-1),0\big) & f\big((N-1),1\big) & \cdots & f\big((N-1),(M-1)\big) \end{bmatrix}$$

Proses pengambilan citra digital dilakukan melalui *sampling* dan kuantisasi. *Sampling* merupakan proses pemetaan citra menjadi baris dan kolom. Sedangkan kuantisasi merupakan proses menentukan nilai intensitas warna setiap piksel di setiap baris dan kolom yang telah dibuat.

2.1.5 Segmentasi Citra

Segmentasi citra merupakan proses pemisahan objek yang diinginkan dari objek lain disekitarnya (Andono, dkk., 2017). Proses segmentasi dilakukan untuk memudahkan proses pengenalan pola pada objek yang diinginkan pada citra

tersebut. Dalam proses segmentasi, pada umumnya objek lain di sekitar citra akan dihilangkan dengan mengubah nilai intensitasnya menjadi 0.

Pada penelitian ini, proses segmentasi dilakukan dengan menggunakan algoritma k-means. K-means merupakan algoritma yang dapat digunakan untuk membagi n data menjadi k bagian $(k \leq n)$ berdasarkan nilai rata-rata (means) terdekat dari masing-masing data (Arsy, 2016). Secara matematis, algoritma k-means dapat dituliskan dalam Persamaan (2.15).

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 \tag{2.15}$$

dimana $S = (S_1, S_2, \dots, S_k)$ merupakan himpunan bagian/kelompok, x merupakan data pada S_i, dan μ_i merupakan nilai rata-rata x pada kelompok S_i. Nilai rata-rata μ_i dapat ditulis dalam Persamaan (2.16).

$$\mu_i = \frac{1}{|S_i|} \sum_{x \in S_i} x \tag{2.16}$$

dimana $|S_i|$ merupakan banyak data pada kelompok S_i.

Pada penelitian ini, algoritma k-means dilakukan terhadap citra dalam bentuk HSV (*Hue*, *Saturation*, *Value*). Hal ini dilakukan untuk mendapatkan hasil segmentasi terbaik. Adapun proses pengubahan citra RGB menjadi HSV dapat dilakukan dengan menggunakan nilai minimum dan maksimum intensitas RGB yang telah dinormalisasi (Chernov, 2015). Secara matematis, hal ini dapat dituliskan dalam Persamaan (2.17), (2.18), (2.19), (2.20), dan (2.21).

$$R' = \frac{R}{255} \tag{2.17}$$

$$G' = \frac{G}{255} \tag{2.18}$$

$$B' = \frac{B}{255} \tag{2.19}$$

$$Cmax = max\ (R', G', B') \tag{2.20}$$

$$Cmin = min\ (R', G', B') \tag{2.21}$$

Selanjutnya, kita dapat memperoleh selisih antara nilai maksimum dan minimum tersebut seperti ditulis pada Persamaan (2.22).

$$\Delta = Cmax - Cmin \tag{2.22}$$

Dengan menggunakan persamaan-persamaan di atas, selanjutnya kita dapat mencari nilai HSV masing-masing menggunakan Persamaan (2.23), (2.24), dan (2.25).

$$H = \begin{cases} \frac{1}{6} \times \left(\frac{G' - B'}{\Delta}\ mod\ 6 \right) & , Cmax = R' \\ \frac{1}{6} \times \left(\frac{B' - R'}{\Delta} + 2 \right) & , Cmax = G' \\ \frac{1}{6} \times \left(\frac{R' - G'}{\Delta} + 4 \right) & , Cmax = B' \end{cases} \tag{2.23}$$

$$S = \begin{cases} 0 & , Cmax = 0 \\ \frac{\Delta}{Cmax} & , Cmax \neq 0 \end{cases} \tag{2.24}$$

$$V = Cmax \tag{2.25}$$

Setelah citra disegmentasi menggunakan k-means, selanjutnya objek rimpang akan dipisahkan melalui proses *thresholding*. *Thresholding* merupakan salah satu metode segmentasi citra yang memisahkan antara objek dengan background dalam suatu citra berdasarkan pada perbedaan intensitas cahayanya (Pamungkas, 2017).

Pada penelitian ini, metode *thresholding* yang digunakan adalah metode Otsu. Metode Otsu merupakan metode *thresholding* yang dilakukan dengan

meminimalkan nilai varians dalam kelas pada histogram citra (Yousef, 2011). Varians dalam kelas dapat didefinisikan dengan Persamaan (2.26).

$$\sigma_{\omega}^2 = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \tag{2.26}$$

dimana $\omega_0(t)$ dan $\omega_1(t)$ merupakan nilai probabilitas masing-masing kelas yang dipisahkan terhadap nilai t, sedangkan $\sigma_0(t)$ dan $\sigma_1(t)$ merupakan nilai varians masing-masing kelas.

Peluang $\omega_0(t)$ dan $\omega_1(t)$ dapat ditulis dalam Persamaan (2.27) dan Persamaan (2.28).

$$\omega_0(t) = \sum_{i=0}^{t-1} p(i) \tag{2.27}$$

$$\omega_1(t) = \sum_{i=t}^{L-1} p(i) \tag{2.28}$$

dimana t merupakan nilai ambang yang digunakan untuk memisahkan masing-masing kelas, sedangkan nilai L merupakan banyak nilai intensitas yang mungkin. Nilai $p(i)$ merupakan nilai peluang munculnya intensitas i pada citra.

Varians masing-masing kelas dapat didefinisikan dalam Persamaan (2.29) dan Persamaan (2.30).

$$\sigma_0(t) = \sum_{i=0}^{t-1} \big(i - \mu_0(t)\big)^2 \frac{p(i)}{\omega_0(t)} \tag{2.29}$$

$$\sigma_1(t) = \sum_{i=t}^{L-1} \big(i - \mu_1(t)\big)^2 \frac{p(i)}{\omega_1(t)} \tag{2.30}$$

dimana $\mu_0(t)$ dan $\mu_1(t)$ merupakan nilai rata-rata dari masing-masing kelas. Nilai rata-rata masing-masing kelas dapat ditulis dalam Persamaan (2.31) dan (2.32).

$$\mu_0(t) = \sum_{i=0}^{t-1} i \frac{p(i)}{\omega_1(t)} \tag{2.31}$$

$$\mu_1(t) = \sum_{i=t}^{L-1} i \frac{p(i)}{\omega_1(t)} \tag{2.32}$$

2.1.6 Ekstraksi Fitur Warna

Ekstraksi fitur warna merupakan proses pengambilan warna dominan dari objek sebagai salah satu ciri/fitur dari objek tersebut. Pada penelitian ini, proses ekstraksi fitur warna dilakukan dengan mencari rata-rata dari setiap fitur warna pada citra yaitu fitur warna merah, hijau, dan biru. Secara matematis, proses ekstraksi fitur warna objek pada citra dapat dirumuskan dalam Persamaan (2.33).

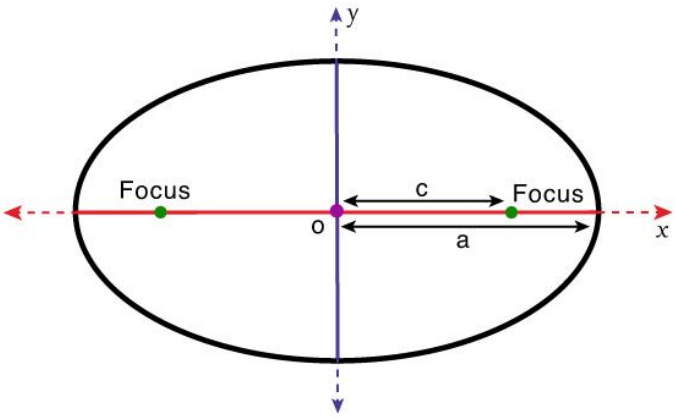
$$f(z) = \frac{1}{N} \sum I(x, y, z) \tag{2.33}$$

dimana N merupakan banyak piksel pada objek, x dan y merupakan koordinat setiap piksel pada objek, dan z merepresentasikan jenis fitur warna yang diekstraksi. Dengan demikian, proses ekstraksi fitur warna akan menghasilkan tiga nilai yang merepresentasikan warna dominan dari objek.

2.1.7 Ekstraksi Fitur Bentuk

Ekstraksi fitur bentuk digunakan untuk membedakan bentuk satu objek dengan bentuk objek lainnya. Pada pengenalan pola objek pada citra, fitur bentuk yang dapat digunakan adalah *eccentricity* dan *metric* (Rachmawanto, 2018). Ekstraksi fitur suatu objek pada citra dapat dilakukan dengan memisalkan objek tersebut sebagai elips.

Pada elips, *eccentricity* merupakan perbandingan antara jarak fokus elips terkecil (c) dengan jarak fokus elips terbesar (a) Dapat dilihat pada Gambar 2.8.



Gambar 2.8 *Eccentricity* Dari Sebuah Elips
Sumber: Mathmonks, 2023

Dengan demikian, *eccentricity* dapat dituliskan dalam Persamaan (2.34).

$$e = \frac{c}{a} = \frac{\sqrt{a^2 - b^2}}{a} = \sqrt{1 - \frac{b^2}{a^2}} \tag{2.34}$$

dimana b merupakan panjang sumbu elips terkecil dan a merupakan panjang sumbu elips terbesar yang juga sekaligus jarak fokus elips terbesar. *Eccentricity* memiliki kisar nilai antara 0-1.

Metric merupakan rasio antara *area* (luas) dan *perimeter* (keliling) elips. *Metric* dalam pengenalan pola objek juga disebut sebagai tingkat kebulatan. Nilai *metric* juga berkisar antara 0-1. Apabila nilai *metric* dari suatu objek semakin mendekati angka 1, maka bentuk objek tersebut akan semakin mendekati bentuk lingkaran. Sedangkan apabila nilai *metric* dari suatu objek semakin mendekati angka 0, maka bentuk objek tersebut akan semakin jauh dari bentuk lingkaran dan akan semakin mendekat terhadap bentuk garis. Secara matematis, *metric* dapat dihitung melalui Persamaan (2.35).

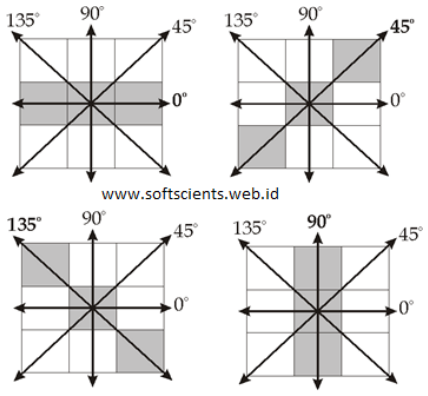
$$M = \frac{4\pi A}{P^2} \tag{2.35}$$

dimana M adalah *metric*, A adalah *area*, dan P adalah *perimeter*.

Dalam pengolahan citra, *area* juga dapat direpresentasikan sebagai banyak piksel dari suatu objek pada citra. Sedangkan *perimeter* dapat direpresentasikan sebagai banyak piksel yang berada di batas objek tersebut.

2.1.8 Ekstraksi Fitur Tekstur

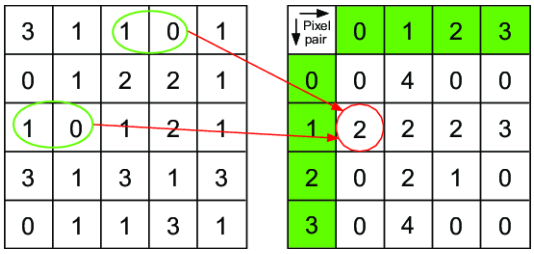
Pada penelitian ini, ekstraksi fitur tekstur dilakukan dengan menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM). GLCM menggunakan matriks untuk merepresentasikan hubungan antara dua piksel yang bertetanggaan dengan sudut θ (Yunus, 2020). Pada penelitian ini, sudut yang digunakan untuk mempelajari tekstur citra adalah sudut $0°$, $45°$, $90°$, dan $135°$. Keempat sudut tersebut sudah mewakili keseluruhan hubungan antar dua piksel pada citra seperti terlihat pada Gambar 2.9.



Gambar 2.9 Sudut Yang Dapat Digunakan Pada GLCM
Sumber: Softscients, 2021

GLCM merupakan matriks berukuran $N \times N$ dimana N merupakan banyak intensitas yang mungkin. Pada citra, nilai intensitas yang mungkin adalah 0-255

sehingga banyak intensitas yang mungkin adalah 256. Baris pada GLCM merepresentasikan nilai intensitas piksel pertama, sedangkan kolom pada GLCM merepresentasikan nilai intensitas piksel kedua. Setiap elemen pada GLCM merepresentasikan banyak hubungan antar dua piksel yang ada pada citra seperti terlihat pada Gambar 2.10.



Gambar 2.10 Proses Perhitungan Jumlah Hubungan Dua Piksel
Sumber: Verma, 2015

Setelah proses perhitungan setiap hubungan antar dua piksel pada citra selesai, maka proses selanjutnya adalah menambahkan matriks GLCM dengan hasil *transpose* matriksnya sehingga diperoleh matriks simetris. Selanjutnya, matriks GLCM dinormalisasi dengan membagi setiap elemen matriks GLCM dengan jumlah seluruh nilai elemen matriks GLCM. Secara matematis, proses normalisasi matriks GLCM dapat dituliskan dalam Persamaan (2.36).

$$glcmNorm = \frac{glcmValue}{\sum_i^N glcmValue} \tag{2.36}$$

Selanjutnya, dengan menggunakan matriks GLCM dapat diperoleh beberapa parameter yang merepresentasikan fitur tekstur. Pada penelitian ini, digunakan enam parameter sebagai fitur tekstur yaitu *dissimilarity*, *correlation*, *homogeneity*, *contrast*, ASM, dan *energy* (Softscients, 2021).

1. *Dissimilarity*

   Dissimilarity merupakan parameter yang digunakan untuk mengukur tingkat ketidakmiripan suatu tekstur. Semakin acak tekstur suatu citra, maka

akan semakin tinggi nilai *dissimilarity* pada citra tersebut. *Dissimilarity* dapat dihitung menggunakan Persamaan (2.37).

$$dissimilarity = \sum_{i}^{N}\sum_{j}^{N}|i-j|P(i,j) \tag{2.37}$$

dimana $P(i,j)$ merupakan matriks GLCM dengan koordinat (i,j).

2. *Correlation*

Correlation merupakan parameter yang mengukur hubungan linear intensitas piksel ketetanggaan. Secara matematis, *correlation* dapat dituliskan dalam Persamaan (2.38), (2.39), (2.40), (2.41), dan (2.42).

$$correlation = \frac{\sum_{i}^{N}\sum_{j}^{N}(i-\mu_x)(j-\mu_y)P(i,j)}{\sqrt{\delta_x\delta_y}} \tag{2.38}$$

$$\mu_x = \sum_{i}^{N}\sum_{j}^{N} i\,P(i,j) \tag{2.39}$$

$$\mu_y = \sum_{i}^{N}\sum_{j}^{N} j\,P(i,j) \tag{2.40}$$

$$\delta_x = \sum_{i}^{N}\sum_{j}^{N}(i-\mu_x)^2\,P(i,j) \tag{2.41}$$

$$\delta_y = \sum_{i}^{N}\sum_{j}^{N}(j-\mu_x)^2\,P(i,j) \tag{2.42}$$

3. *Homogeneity*

Homogeneity adalah parameter yang mengukur homogenitas atau kesamaan variasi intensitas citra. Nilai *homogeneity* akan bernilai tinggi jika setiap piksel memiliki intensitas yang sama. *Homogeneity* dapat ditulis dalam Persamaan (2.43).

$$homogeneity = \sum_{i}^{N}\sum_{j}^{N}\frac{P(i,j)}{1+(i-j)^2} \tag{2.43}$$

4. *Contrast*

Contrast merupakan parameter yang merepresentasikan variasi perbedaan intensitas antar piksel yang bertetanggan. Semakin sama dan

semakin konstan intensitas antar piksel yang bertetanggaan, maka nilai *contrast* akan semakin mendekati 0. *Contrast* dapat dihitung menggunakan Persamaan (2.44).

$$contrast = \sum_{i}^{N}\sum_{j}^{N}(i-j)^2\,P(i,j) \tag{2.44}$$

5. *Angular Second Moment* (ASM)

Angular Second Moment (ASM) merupakan parameter yang digunakan untuk mengukur tingkat keseragaman intensitas piksel. Secara matematis, ASM dapat dihitung menggunakan Persamaan (2.45).

$$ASM = \sum_{i}^{N}\sum_{j}^{N}P(i,j)^2 \tag{2.45}$$

6. *Energy*

Energy merupakan hasil akar dari ASM. *Energy* juga digunakan untuk mengukur tingkat keseragaman piksel pada citra. Secara matematis, *energy* dapat ditulis dalam Persamaan (2.46).

$$Energy = \sqrt{ASM} \tag{2.46}$$

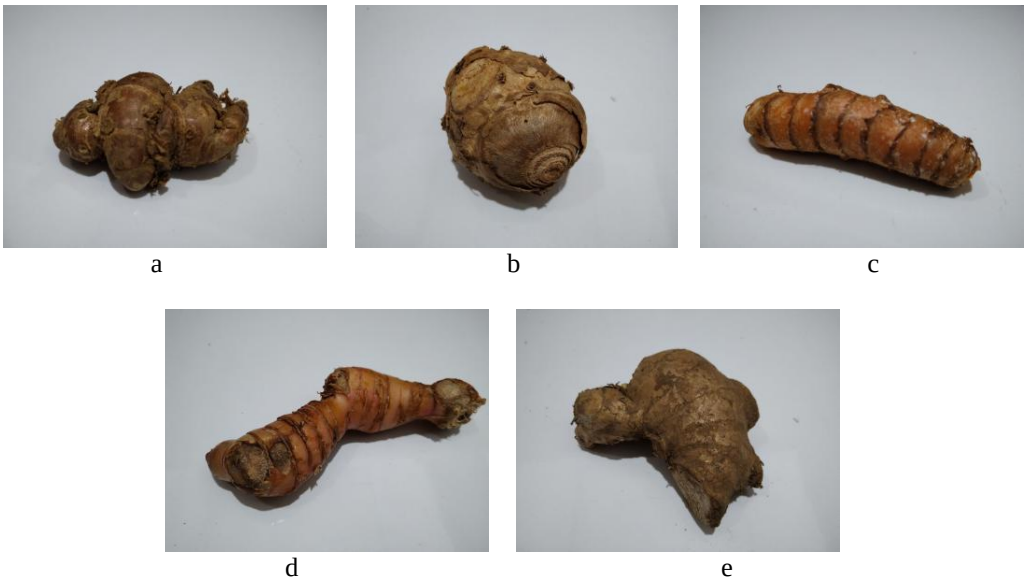
Proses ekstraksi fitur tekstur pada umumnya dilakukan terhadap citra dengan format *grayscale*. Citra RGB dapat diubah menjadi *grayscale* dengan menggunakan Persamaan (2.47).

$$Y = 0{,}299\,R + 0{,}587\,G + 0{,}114\,B \tag{2.47}$$

2.1.9 Rimpang

Rimpang adalah batang tanaman yang merambat di bawah permukaan tanah dan dapat menghasilkan tunas dan akar baru dari ruas-ruasnya (Rizki, 2022).

Tanaman rimpang berkembang biak secara vegetatif alami. Tanaman rimpang memiliki ciri-ciri batangnya beruas, berbuku buku dan bertunas. Contoh tanaman yang tergolong rimpang antara lain kencur, temulawak, kunyit, lengkuas, jahe.



a b c d e

Gambar 2.11 Jenis-Jenis Tanaman Rimpang (a) Kencur (b) Temulawak (c) Kunyit (d) Lengkuas (e) Jahe

1. Kencur

   Kencur memiliki rimpang pendek yang bercabang dan bergerombol di bawah tanah dan sebagian tumbuh di permukaan tanah seperti terlihat pada Gambar 2.11 (a). Kulit coklat terang sedangkan bagian dalamnya kuning kencur memiliki tekstur yang lunak/mudah dihancurkan, tidak berserat dan mengandung cukup banyak air.

2. Temulawak

   Temulawak merupakan tanaman rimpang yang memiliki bentuk cenderung bulat dan memiliki bintik-bintik hitam di permukaan kulitnya seperti terlihat pada Gambar 2.11 (b). Pada umumnya, temulawak memiliki ukuran yang lebih besar daripada kunyit. Warna dari temulawak adalah coklat gelap.

3. Kunyit

Kunyit memiliki batang yang tumbuh di bawah permukaan tanah. Rimpang kunyit berbentuk silindris dan ujung meruncing seperti terlhiat pada Gambar 2.11 (c). Kulit luarnya jingga kecoklatan, sedangkan bagian dalamnya merah jingga kekuningan. Daunnya kuning kehijauan dan lebar. Kunyit memiliki tekstur yang lembut sehingga mudah dipotong dan diparut.

4. Lengkuas

Lengkuas memiliki batang yang dipenuhi dengan pelepah daun. Rimpang lengkuas merupakan rimpang yang besar, berdaging, bercabang, dan silindris. Kulit luarnya coklat kemerahan atau kuning kehijauan pucat, sisik putih atau kemerahan sedangkan bagian dalamnya putih seperti terlihat pada Gambar 2.11 (d). Lengkuas memiliki kulit luar yang licin dan mengkilat namun tekstur bagian dalamnya kasar dan berserat. Lengkuas memiliki alur ruas yang jelas karena bentuk seratnya yang Panjang dan kokoh.

5. Jahe

Jahe memiliki rimpang berbentuk jari yang melengkung di bagian ruas tengah seperti terlihat pada Gambar 2.11 (e). Rimpang jahe berwarna kuning hingga kemerahan. Jahe memiliki tekstur yang keras dan berserabut.

2.2 Kajian Integrasi Topik dengan Al-Qur’an/Hadits

Dalam Al-Qur’an telah dijelaskan bahwasanya Allah telah menciptakan bumi dengan beragam jenis dan warna di dalamnya. Hal ini merupakan salah satu

bentuk kebesaran Allah SWT sebagaimana tertulis dalam Al-Qur'an surat An-Nahl ayat 13 yang berbunyi (Kemenag RI, 2022):

وَمَا ذَرَاَ لَكُمْ فِى الْاَرْضِ مُخْتَلِفًا اَلْوَانُهٗ ۗاِنَّ فِيْ ذٰلِكَ لَاٰيَةً لِّقَوْمٍ يَّذَّكَّرُوْنَ

"Dia juga mengendalikan apa yang Dia ciptakan untukmu di bumi ini dengan berbagai jenis dan macam warnanya. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda (kebesaran Allah) bagi kaum yang mengambil pelajaran" (Q.S. An-Nahl:13)

Sebagai manusia, kita diberikan akal oleh Allah SWT untuk dapat mempelajari dan memikirkan tentang segala kebesaran Allah SWT. Dalam ayat tersebut dijelaskan bahwa Allah SWT telah menciptakan segala sesuatu dengan berbagai jenis dan macam.

Rimpang merupakan salah satu ciptaan Allah yang memiliki beragam jenis dan warna yang berbeda. Meskipun sebagian orang tidak bisa membedakan antar jenis tanaman rimpang, pada dasarnya setiap jenis rimpang memiliki karakteristik dan ciri khas yang dapat membedakan satu sama lainnya.

Salah satu teknologi yang dapat dikembangkan untuk membedakan tanaman rimpang adalah *machine learning*. Dengan demikian, pengembangan *machine learning* pada penelitian ini juga merupakan bentuk syukur terhadap ciptaan Allah SWT, yaitu rimpang. Selain itu, penelitian ini juga dapat membantu masyarakat dalam membedakan rimpang dengan bantuan *machine learning*. Hal ini juga merupakan salah satu bentuk kebaikan yang dianjurkan oleh Allah yaitu untuk saling tolong menolong. Hal ini juga bersesuaian dengan hadits sebagai berikut (An-Nawawi,1999).

عَنْ أَبِي هُرَيْرَةَ رَضِيَ اللهُ عَنْهُ، عَنِ النَّبِيِّ ﷺ قَالَ: «مَنْ نَفَّسَ عَنْ مُؤْمِنٍ كُرْبَةً مِنْ كُرَبِ الدُّنْيَا، نَفَّسَ اللهُ عَنهُ كُرْبَةً مِنْ كُرَبِ يَوْمِ القِيَامَةِ. وَمَنْ يَسَّرَ عَلَى مُعْسِرٍ، يَسَّرَ اللهُ عَلَيْهِ فِي

الدُّنْيَا وَالآخِرَةِ. وَمَنْ سَتَرَ مُسْلِماً سَتَرَهُ اللهُ فِي الدُّنْيَا وَالآخِرَةِ. وَاللهُ فِي عَوْنِ العَبْدِ مَا كَانَ العَبْدُ فِي عَوْنِ أَخِيْهِ. وَمَنْ سَلَكَ طَرِيْقاً يَلْتَمِسُ فِيْهِ عِلْماً سَهَّلَ اللهُ لَهُ بِهِ طَرِيْقاً إِلَى الجَنَّةِ. وَمَا اجْتَمَعَ قَوْمٌ فِي بَيْتٍ مِنْ بُيُوْتِ اللهِ يَتْلُوْنَ كِتَابَ اللهِ وَيَتَدَارَسُوْنَهُ بَيْنَهُمْ إِلَّا نَزَلَتْ عَلَيْهِمُ السَّكِيْنَةُ، وَغَشِيَتْهُمُ الرَّحْمَةُ، وَحَفَّتْهُمُ الْمَلَائِكَةُ، وَذَكَرَهُمُ اللهُ فِيْمَنْ عِنْدَهُ، وَمَنْ بَطَّأَ بِهِ عَمَلُهُ لَمْ يُسْرِعْ بِهِ نَسَبُهُ» رَوَاهُ مُسْلِمٌ بِهَذَا اللَّفْظِ.

"Dari Abu Hurairah radhiyallahu 'anhu, dari Nabi shallallahu 'alaihi wa sallam bersabda, "Barangsiapa yang menghilangkan kesusahan dari kesusahan-kesusahan dunia orang mukmin, maka Allah akan menghilangkan kesusahan dari kesusahan-kesusahan hari kiamat. Barangsiapa yang memberi kemudahan orang yang kesulitan (utang), maka Allah akan memberi kemudahan baginya di dunia dan akhirat. Siapa yang menutup aib seorang muslim, maka Allah akan menutup aibnya di dunia dan di akhirat. Siapa saja yang menolong saudaranya, maka Allah akan menolongnya sebagaimana ia menolong saudaraya. Barangsiapa yang menempuh perjalanan dalam rangka menuntut ilmu, maka Allah akan mudahkan baginya jalan menuju surga. Tidaklah berkumpul sekelompok orang di salah satu rumah Allah (masjid) untuk membaca Kitabullah dan saling mempelajarinya di antara mereka, melainkan akan turun kepada mereka ketenangan, rahmat meliputinya, para malaikat mengelilinginya, dan Allah menyanjung namanya kepada Malaikat yang ada di sisi-Nya. Barangsiapa yang lambat amalnya, maka tidak akan bisa dikejar oleh nasabnya (garis keturunannya yang mulia)." (H.R. Muslim)

Hadits tersebut menjelaskan bahwa Allah akan meringankan kesulitan bagi setiap manusia yang membantu meringankan kesulitan saudaranya. Dengan demikian, penelitian ini merupakan implementasi dari sifat tolong menolong yang telah diajarkan dalam agama Islam.

2.3 Kajian Topik dengan Teori Pendukung

Objektif dari penelitian ini adalah untuk mengembangkan ilmu pengetahuan guna membantu masyarakat dalam membedakan jenis tanaman rimpang. Tanaman rimpang cenderung memiliki warna, tekstur, dan bentuk yang hampir mirip antar satu sama lainnya. Hal ini menyebabkan sulitnya masyarakat dalam membedakan antar jenis tanaman rimpang. Setiap tanaman rimpang memiliki fungsi dan

kegunaan masing-masing. Tanaman rimpang sering digunakan sebagai bahan makanan maupun obat-obatan herbal. Oleh karena itu, membedakan tanaman rimpang merupakan hal yang penting untuk dipelajari guna membantu manusia dalam membuat makanan atau obat-obatan sesuai dengan manfaatnya.

Dengan menggunakan teknologi masa kini, tanaman rimpang dapat dibedakan melalui proses klasifikasi. Proses klasifikasi merupakan proses pengelompokkan data berdasarkan kelasnya. Pada penelitian ini, dikembangkan model *machine learning* sebagai alat untuk klasifikasi jenis tanaman rimpang berdasarkan warna, bentuk, dan tekstur. Data yang digunakan dalam penelitian ini merupakan data rempah yang diambil dari *website* Kaggle dengan total 100 data latih dan 25 data uji.

Proses klasifikasi pada penelitian ini dibagi menjadi beberapa tahapan. Tahapan pertama yang perlu dilakukan adalah tahap *preprocessing* citra. Tahap *preprocessing* dilakukan untuk menyiapkan citra guna memaksimalkan hasil klasifikasi. Tahap *preprocessing* yang dilakukan antara lain adalah segmentasi.

Setelah *preprocessing*, tahap kedua yang dilakukan adalah tahap ekstraksi fitur. Tahap ekstraksi fitur dilakukan untuk mempelajari dan mendapatkan karakteristik objek tanaman rimpang. Terdapat tiga fitur yang akan diekstraksi antara lain yaitu fitur warna, bentuk dan tekstur. Ekstraksi fitur warna dilakukan dengan mencari warna dominan di setiap kanal baik kanal merah, hijau, maupun biru. Pada tahap ekstraksi fitur bentuk, digunakan nilai *eccentricity* dan *metric* sebagai parameter yang merepresentasikan fitur bentuk. Sedangkan pada tahap ekstraksi fitur tekstur, digunakan enam parameter utama antara lain *dissimilarity*, *correlation*, *homogeneity*, *contrast*, ASM, dan *energy*.

Setelah dilalui tahap *preprocessing* dan tahap ekstraksi fitur, selanjutnya dilakukan klasifikasi jenis tanaman rimpang menggunakan algoritma SVM. SVM merupakan salah satu jenis metode *machine learning* yang umum digunakan untuk klasifikasi dan regresi. Keunggulan dari metode ini adalah dapat menghasilkan model klasifikasi yang baik pada data yang relatif sedikit. Dengan menggunakan metode ini, diharapkan pada penelitian ini dapat dihasilkan hasil klasifikasi dengan tingkat akurasi yang tinggi.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian yang digunakan pada penelitian ini adalah penelitian kuantitatif dengan metode eksperimen. Dengan demikian, penelitian ini dilakukan untuk menguji efektif atau tidaknya variabel eksperimen. Hasil yang digunakan untuk menguji pada penelitian ini adalah hasil berupa tingkat akurasi klasifikasi jenis tanaman rimpang menggunakan algoritma SVM berdasarkan fitur warna, bentuk, dan tekstur.

3.2 Data dan Sumber Data

Data yang digunakan pada penelitian ini yakni data berupa citra rimpang. Data yang digunakan dalam penelitian ini merupakan data rempah yang diperoleh dari dataset di *website* Kaggle. Jenis tanaman rimpang terdapat lima antara lain kencur, temulawak, kunyit, lengkuas, dan jahe. Jumlah citra yang digunakan sebanyak 125 citra dengan pembagian data latih sebesar 80% dan data uji sebesar 20%. Dengan demikian, jumlah data latih yang digunakan adalah 100 citra dan jumlah data uji yang digunakan adalah 25 citra dimana masing-masing jenis rimpang akan diambil 20 citra sebagai data latih dan lima citra sebagai data uji. Masing-masing citra memiliki ukuran dimensi 2364×1773.

3.3 Tahapan Penelitian

Adapun tahapan penelitian pada penelitian ini dapat dilihat pada Gambar 3.1 dan diuraikan sebagai berikut.

1. Segmentasi Citra

   Tahap segmentasi citra dilakukan beberapa proses sebagai berikut.

   a. Citra rimpang dengan format RGB diubah ke dalam format HSV terlebih dahulu untuk memaksimalkan proses segmentasi menggunakan Persamaan (2.23), (2.24) dan (2.25).
   b. Citra rimpang yang telah diubah ke dalam format HSV disegmentasi menggunakan algoritma K-means menggunakan Persamaan (2.15) sehingga objek rimpang memiliki warna yang relative berbeda terhadap objek sekitarnya. Nilai k yang digunakan dalam penelitian ini adalah 5.
   c. Hasil segmentasi K-means selanjutnya akan dilakukan *thresholding* sehingga dihasilkan citra biner yang berfungsi sebagai *mask* menggunakan Persamaan (2.26).
   d. Setelah diperoleh citra *mask* selanjutnya citra tersebut akan dikalikan dengan citra rimpang dengan format RGB sehingga dihasilkan citra baru dimana objek selain rimpang akan memiliki nilai piksel $(0,0,0)$.

2. Ekstraksi Fitur

   Proses ekstraksi fitur pada penelitian ini dilakukan berdasarkan fitur warna, bentuk, dan tekstur. Adapun penjelasan masing-masing ekstraksi fitur adalah sebagai berikut.

   a. Ekstraksi fitur warna dilakukan dengan mencari rata-rata masing-masing kanal warna yaitu R, G, dan B menggunakan Persamaan (2.33).

Proses ekstraksi fitur warna menghasilkan vektor fitur dengan ukuran 3×1.

b. Ekstraksi fitur bentuk pada penelitian ini dilakukan berdasarkan nilai *eccentricity* dan *metric* (2.34) dan (2.35). Ekstraksi fitur bentuk menghasilkan vektor fitur dengan ukuran 2×1.

c. Ekstraksi fitur tekstur menghasilkan vektor fitur dengan ukuran 24×1 yang diperoleh dari ekstraksi fitur tekstur berdasarkan *dissimilarity*, *correlation*, *homogeneity*, *contrast*, ASM, dan *energy* yang dihitung terhadap empat sudut yaitu sudut $0°$, $45°$, $90°$, dan $135°$.

3. Klasifikasi Citra

Setelah proses ekstraksi fitur, selanjutnya dilakukan klasifikasi jenis rimpang menggunakan algoritma SVM yang memiliki beberapa tahapan berikut.

a. Membagi data menjadi dua kelas untuk memudahkan proses klasifikasi.

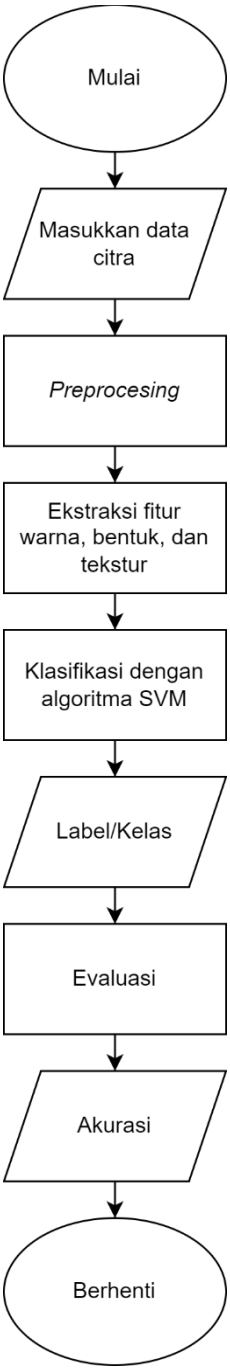
b. Pilih nilai w_0 dan b_0 secara acak dimana w_0 merupakan vektor dengan ukuran 29×1 dan b_0 merupakan nilai konstanta.

c. Lakukan perhitungan menggunakan Persamaan (2.12) dan (2.13) sehingga diperoleh nilai *w* dan *b* selanjutnya

d. Lakukan proses tersebut secara berulang sehingga dihasilkan nilai *w* dan *b* yang optimal.

e. Gunakan nilai *w* dan *b* yang optimal dalam model SVM sehingga diperoleh model SVM pertama.

f. Ulangi langkah 1 hingga 5 sebanyak $\frac{n(n-1)}{2} = \frac{5(5-1)}{2} = 10$ kali sehingga diperoleh 10 model SVM.



Gambar 3.1 Diagram Alir Penelitian

4. Evaluasi

Melakukan tahap evaluasi melalui beberapa tahapan berikut.

a. Bagi data menjadi data latih dan data uji menggunakan algoritma *K-fold Cross Validation* dengan rasio 80% data latih dan 20% data uji.

b. Lakukan proses klasifikasi terhadap masing-masing data yang telah dibagi.
c. Gunakan *confusion matrix* untuk memetakan hasil klasifikasi terhadap masing-masing data uji.
d. Selanjutnya hitung nilai akurasi masing-masing proses klasifikasi menggunakan Persamaan (2.14).
e. Hitung rata-rata nilai akurasi masing-masing proses klasifikasi tersebut sehingga diperoleh nilai akurasi sesungguhnya.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Segmentasi Citra

Proses segmentasi citra pada penelitian ini dilakukan untuk memisahkan objek rimpang dari objek lainnya seperti latar belakang atau bayangan rimpang tersebut. Proses segmentasi dilakukan dengan bantuan algoritma k-means. Algoritma k-means dapat melakukan generalisasi warna sehingga perbedaan warna tiap objek pada citra menjadi lebih kontras sehingga proses pemisahan rimpang dapat dilakukan lebih mudah.

Untuk mengoptimalkan hasil segmentasi, citra yang memiliki format RGB perlu diubah menjadi HSV. Citra HSV memiliki keunggulan dalam menampilkan perbedaan kontras tiap intensitas warna. Dengan mengubah format citra ke dalam format HSV, maka proses segmentasi dapat dilakukan dengan lebih mudah dan akurat. Sebagai contoh proses pengubahan citra RGB ke HSV, kita gunakan salah satu sampel data dapat dilihat pada Gambar 4.1.

Gambar 4.1 Citra Sampel

dimana data tersebut dapat direpresentasikan dalam bentuk matriks sebagai berikut (selengkapnya dapat dilihat pada Lampiran 1).

$$\begin{bmatrix} (157,158,162) & \cdots & (160,160,162) \\ \vdots & \ddots & \vdots \\ (167,172,172) & \cdots & (164,168,171) \end{bmatrix}$$

Selanjutnya, dengan menggunakan Persamaan (2.17), (2.18), dan (2.19). Maka dapat ditemukan nilai R', G', dan B' dari titik piksel pada baris dan kolom pertama pada Persamaan (4.1), (4.2), dan (4.3).

$$R' = \frac{157}{255} = 0{,}615 \tag{4.1}$$

$$G' = \frac{158}{255} = 0{,}619 \tag{4.2}$$

$$B' = \frac{162}{255} = 0{,}635 \tag{4.3}$$

Dengan demikian, dapat diperoleh nilai $Cmax$ dan $Cmin$ berturut-turut dengan menggunakan Persamaan (2.20) dan (2.21) yaitu sebesar $0{,}635$ dan $0{,}615$. Selanjutnya dengan menggunakan Persamaan (2.22) diperoleh $\Delta = 0{,}02$.

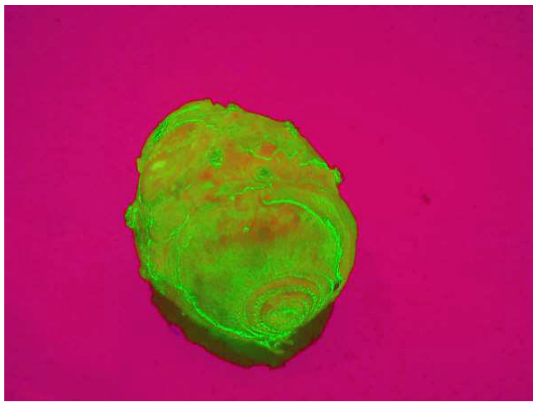
Dengan menggunakan nilai $Cmax$, $Cmin$, dan Δ yang telah dihitung, maka dengan menggunakan Persamaan (2.23), (2.24), (2.25) dapat diperoleh hasil pada Persamaan (4.4), (4.5), dan (4.6)

$$H = \frac{1}{6} \times \left(\frac{0{,}615 - 0{,}619}{0{,}02} + 4 \right) = 0{,}634 \tag{4.4}$$

$$S = \frac{0{,}02}{0{,}635} = 0{,}031 \tag{4.5}$$

$$V = 0{,}635 \tag{4.6}$$

Dengan demikian, bentuk HSV dari piksel baris pertama dan kolom pertama pada sampel tersebut adalah (0,634; 0,031; 0,635). Langkah-langkah ini diterapkan terhadap seluruh piksel pada matriks citra sampel. Setelah itu, seluruh piksel pada matriks dikalikan dengan 255 untuk keperluan visualisasi citra. Hasil proses pengubahan citra RGB menjadi HSV dapat dilihat pada Gambar 4.2.

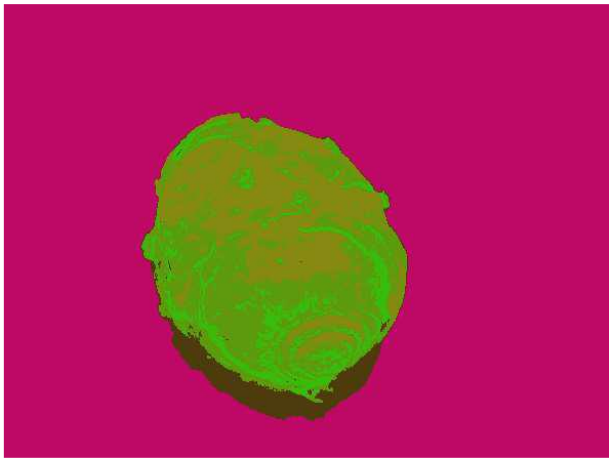


Gambar 4.2 Citra Rimpang Dalam Format HSV

Selanjutnya citra HSV disegmentasi menggunakan algoritma k-means. Algoritma k-means akan dilakukan sebanyak n iterasi dengan menggunakan Persamaan (2.15). Pada penelitian ini, nilai k yang digunakan adalah 5. Nilai $k = 5$ dipilih berdasarkan hasil segmentasi terbaik dari eksperimen segmentasi citra. Penggunaan algoritma k-means diharapkan dapat mengelompokkan setiap intensitas piksel pada objek rimpang menjadi satu kesatuan intensitas yang sama.

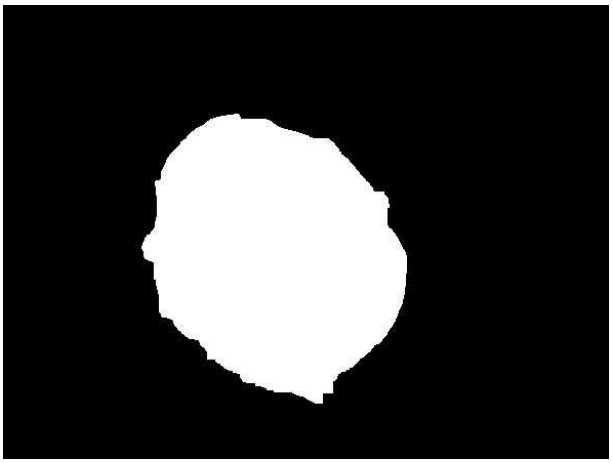
Pada mulanya, algoritma k-means akan memilih lima titik secara acak untuk diasumsikan sebagai titik rata-rata (μ_i) masing-masing kelompok (S_i). Setelah dipilih lima titik tersebut, selanjutnya dihitung jarak setiap piksel terhadap masing-masing μ_i dan mengkategorikan setiap piksel sebagai anggota dalam satu kelompok yang memiliki jarak paling minimum menggunakan Persamaan (2.15). Setelah setiap piksel dikelompokkan terhadap masing-masing kelas, selanjutnya titik rata-rata masing-masing kelompok (μ_i) akan diperbarui menggunakan Persamaan (2.16). Setelah itu, langkah ini diulangi secara terus menerus hingga mencapai hasil paling optimal, yaitu ketika tidak terjadi perubahan signifikan nilai rata-rata masing-masing kelompok. Dengan kata lain, masing-masing titik telah dikelompokkan dengan sesuai. Selanjutnya, setiap piksel yang berada pada

kelompok yang sama akan diberikan nilai yang sama pula berdasarkan nilai rata-rata kelompoknya. Hasil proses segmentasi k-means dapat dilihat pada Gambar 4.3.



Gambar 4.3 Hasil Segmentasi K-means

Hasil segmentasi k-means selanjutnya akan dilakukan proses *thresholding* sehingga diperoleh citra biner yang memisahkan objek rimpang terhadap latar belakang maupun bayangannya. Proses *thresholding* dilakukan dengan mengunakan metode Otsu seperti tertulis secara matematis pada Persamaan (2.26). Setiap piksel objek rimpang pada citra ini akan diberikan nilai intensitas 1. Sedangkan, piksel selain objek rimpang akan diberi nilai intensitas 0. Hasil proses *thresholding* dapat dilihat pada Gambar 4.4.



Gambar 4.4 Citra Hasil *Thresholding*

Citra hasil *thresholding* ini selanjutnya akan dikalikan dengan citra aslinya yang memiliki format RGB. Hasil perkalian kedua citra ini akan menghasilkan citra baru yang hanya memiliki objek rimpang di dalamnya. Setelah itu, citra akan dipangkas untuk meminimalkan banyak piksel yang tidak diperlukan (piksel yang bernilai 0) sehingga diperoleh citra seperti terlihat pada Gambar 4.5. Citra baru ini merupakan hasil akhir proses segmentasi citra pada penelitian ini.

Gambar 4.5 Hasil Segmentasi Citra Rimpang

4.2 Ekstraksi Fitur

Pada penelitian ini, ekstraksi fitur dilakukan untuk mendapatkan nilai karakteristik objek rimpang berdasarkan fitur warna, bentuk, dan tekstur. Fitur warna menghasilkan sebanyak tiga fitur yang terdiri dari fitur warna merah, hijau, dan biru. Fitur bentuk menghasilkan dua fitur yang merepresentasikan nilai *eccentricity* dan *metric* dari objek rimpang. Sedangkan fitur tekstur menghasilkan 24 fitur yang mewakili enam fitur tekstur yaitu *dissimilarity*, *correlation*, *homogeneity*, *contrast*, ASM, dan *energy* terhadap empat sudut yaitu 0°, 45°, 90°, dan 135°. Dengan demikian, fitur total yang dihasilkan dari ekstraksi fitur rimpang

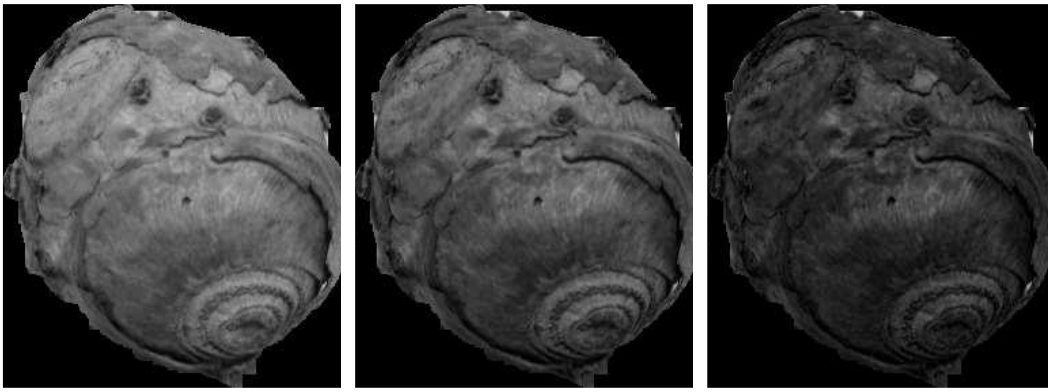
pada penelitian ini adalah 29 fitur. Berdasarkan citra sampel, fitur yang dihasilkan dari objek rimpang tersebut dapat ditulis sebagai vector berukuran 29×1 berikut.

$$\begin{bmatrix} 105 \\ 74 \\ 44 \\ 0{,}553 \\ 0{,}755 \\ 12{,}678 \\ 13{,}909 \\ 13{,}628 \\ 14{,}375 \\ 0{,}867 \\ 0{,}845 \\ 0{,}853 \\ 0{,}836 \\ 0{,}323 \\ 0{,}310 \\ 0{,}318 \\ 0{,}306 \\ 497{,}731 \\ 572{,}652 \\ 548{,}958 \\ 608{,}293 \\ 0{,}071 \\ 0{,}067 \\ 0{,}071 \\ 0{,}065 \\ 0{,}267 \\ 0{,}258 \\ 0{,}267 \\ 0{,}256 \end{bmatrix}$$

4.2.1 Ekstraksi Fitur Warna

Proses ekstraksi fitur warna pada penelitian ini diperoleh dengan menghitung nilai rata-rata intensitas dari seluruh piksel pada objek rimpang dengan menggunakan Persamaan (2.33). Nilai N diperoleh dari banyak piksel yang membentuk objek rimpang pada citra. Untuk memudahkan proses perhitungan fitur

warna, setiap fitur warna baik warna merah, hijau, dan biru dipisahkan terlebih dahulu seperti pada Gambar 4.6.



(a) (b) (c)

Gambar 4.6 Hasil Pemisahan Fitur RGB

Selanjutnya, dilakukan perhitungan rata-rata nilai intensitas pada masing-masing fitur warna menggunakan citra yang telah dipisah. Berdasarkan citra sampel, data dicari rata-rata masing-masing fitur seperti perhitungan pada Persamaan (4.7), (4.8), dan (4.9).

$$f(R) = \frac{1}{51616}(110 + 89 + \cdots + 81) = 104{,}834 \tag{4.7}$$

$$f(G) = \frac{1}{51616}(88 + 66 + \cdots + 63) = 74{,}355 \tag{4.8}$$

$$f(B) = \frac{1}{51616}(66 + 45 + \cdots + 46) = 43{,}81 \tag{4.9}$$

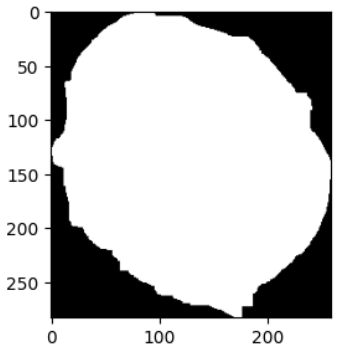
dimana nilai $N = 51616$ diperoleh dengan menghitung banyak piksel yang memiliki nilai intenstias lebih dari 0.

Dengan demikian, proses ekstraksi fitur warna menghasilkan tiga nilai fitur yaitu rata-rata nilai intensitas warna merah sebanyak $104{,}834$, hijau sebanyak $74{,}355$, dan biru sebanyak $43{,}81$. Dengan menggunakan pembulatan, maka diperoleh fitur warna pada citra sampel adalah sebagai berikut.

$$\begin{bmatrix} 105 \\ 74 \\ 44 \end{bmatrix}$$

4.2.2 Ekstraksi Fitur Bentuk

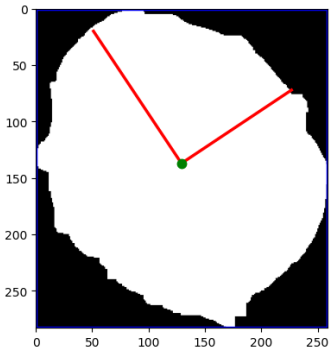
Fitur bentuk yang digunakan pada penelitian ini adalah *eccentricity* dan *metric*. Pada dasarnya, setiap rimpang memiliki kecenderungan bentuk yang berbeda. Oleh karena itu, kedua fitur ini digunakan untuk mempelajari perbedaan masing-masing bentuk rimpang. Baik fitur *eccentricity* maupun *metric* dapat memberi informasi bentuk rimpang. Jika nilai *metric* semakin mendekati 1, maka objek rimpang tersebut memiliki bentuk yang cenderung bulat. Sedangkan jika nilai *metric* semakin mendekati 0, maka bentuk rimpang cenderung menjauhi bentuk bulat. Sedangkan *eccentricity* memiliki sifat yang berkebalikan dari *metric*. Semakin kecil *eccentricity*, maka objek semakin mendekati bentuk lingkaran, sedangkan semakin besar *eccentricity* maka bentuk objek semakin menjauh dari lingkaran. Sebagai contoh, digunakan bentuk biner dari citra sampel yang dapat dilihat pada Gambar 4.7.



Gambar 4.7 Citra Sampel Biner

Untuk mencari nilai *eccentricity* diperlukan jarak fokus elips terbesar (a) dan fokus elips terkecil (c). Untuk mencari kedua jarak tersebut, maka perlu

ditentukan terlebih dahulu titik tengah dari objek rimpang. Pada citra sampel, ditentukan titik tengah berada pada piksel dengan koordinat $(137, 129)$. Selanjutnya dicari jarak titik tengah terhadap setiap piksel yang mengelilingi rimpang. Pada citra sampel, diperoleh jarak fokus elips terbesar yaitu $a = 281{,}573$ dan jarak fokus elips terkecil yaitu $b = 234{,}483$. Visualisasi titik tengah, jarak fokus elips terbesar, dan jarak fokus elips terkecil dapat dilihat pada Gambar 4.8.



Gambar 4.8 Visualisasi *Eccentricity*

Selanjutnya, nilai *eccentricity* dapat dihitung menggunakan Persamaan (2.34) sehingga diperoleh hasil pada Persamaan (4.10).

$$e = \sqrt{1 - \frac{234{,}483^2}{281{,}573^2}} = 0{,}553 \tag{4.10}$$

Nilai *metric* dihasilkan dengan menggunakan Persamaan (2.35) dengan menggunakan nilai *area* dan *perimeter*. Dalam penelitian ini, nilai *area* diperoleh dengan menghitung banyak piksel yang memenuhi objek rimpang pada citra, sedangkan nilai *perimeter* diperoleh dengan menghitung banyak piksel yang mengelilingi objek rimpang pada citra. Pada citra sampel diketahui nilai *area* yaitu 51616 dan *perimeter* yaitu $926{,}909$. Dengan demikian diperoleh nilai *metric* citra sampel tertulis pada Persamaan (4.11).

$$M = \frac{4\pi\ 51616}{926{,}909^2} = 0{,}755 \tag{4.11}$$

Maka, diperoleh fitur bentuk dari citra sampel pada penelitian ini berikut.

$$\begin{bmatrix} 0{,}553 \\ 0{,}755 \end{bmatrix}$$

.

4.2.3 Ekstraksi Fitur Tekstur

Proses ekstraksi fitur tekstur dilakukan menggunakan metode GLCM (*Gray Level Co-occurrence Matrix*). Jumlah hubungan intensitas setiap piksel pada citra akan dihitung berdasarkan sudut tertentu sehingga diperoleh matriks GLCM. Setelah diperoleh matriks GLCM, selanjutnya dihitung enam fitur tekstur yaitu *dissimilarity*, *correlation*, *homogeneity*, *contrast*, ASM, dan *energy*.

Sebelum dilakukan perhitungan matriks GLCM, citra terlebih dahulu diubah menjadi *grayscale* menggunakan persamaan (2.47). Sebagai contoh menggunakan citra sampel, dapat diperoleh nilai piksel pada suatu koordinat pada citra sampel dalam bentuk *grayscale* pada persamaan (4.12).

$$Y = 0{,}299\ (110) + 0{,}587\ (88) + 0{,}114\ (66) = 92{,}07 \tag{4.12}$$

Dengan dilakukan pembulatan, maka diperoleh nilai piksel berbentuk RGB $(110, 88, 66)$ adalah 92 dalam bentuk *grayscale*. Hal ini dilakukan terhadap keseluruhan piksel pada citra sampel. Dengan demikian, diperoleh bentuk matriks citra sampel sebagai berikut (selengkapnya dapat dilihat pada Lampiran 2).

$$\begin{bmatrix} 0 & 0 & \cdots & 70 & 108 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 69 & 63 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 25 & 26 & \cdots & 82 & 80 & \cdots & 0 & 0 \\ 28 & 35 & \cdots & 77 & 76 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Pada penelitian ini, proses ekstraksi fitur GLCM dilakukan sebanyak empat kali menggunakan empat sudut berbeda, yaitu sudut 0°, 45°, 90°, dan 135°. Hal ini dilakukan untuk memperoleh karakteristik tekstur rimpang secara merata dengan sudut pandang dari seluruh arah yang mungkin. Sebagai contoh menggunakan citra sampel, dapat diperoleh matriks GLCM menggunakan sudut 0° sebagai berikut (selengkapnya dapat dilihat pada Lampiran 3).

$$P = \begin{bmatrix} 21136 & 0 & \cdots & 3 & 2 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 10 & 0 & \cdots & 32 & 37 & \cdots & 0 & 0 \\ 3 & 0 & \cdots & 32 & 45 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Setelah matriks GLCM diperoleh, selanjutnya matriks GLCM akan ditambahkan dengan matriks transposenya sehingga diperoleh matriks GLCM yang simetris sebagai berikut (selengkapnya dapat dilihat pada Lampiran 4).

$$P = \begin{bmatrix} 42272 & 0 & \cdots & 13 & 5 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 13 & 0 & \cdots & 64 & 69 & \cdots & 0 & 0 \\ 5 & 0 & \cdots & 69 & 90 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Kemudian, masing-masing elemen matriks GLCM dinormalisasi dengan membagi setiap elemen terhadap jumlah keseluruhan elemennya sehingga diperoleh matriks GLCM berikut (selengkapnya dapat dilihat pada Lampiran 5).

$$P = \begin{bmatrix} 0.28948 & 0 & \cdots & 0{,}00009 & 0{,}00003 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0{,}00009 & 0 & \cdots & 0{,}00044 & 0{,}00047 & \cdots & 0 & 0 \\ 0{,}00003 & 0 & \cdots & 0{,}00047 & 0{,}00062 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Setelah diperoleh matriks GLCM, selanjutnya ekstraksi fitur dapat dilakukan dengan menghitung masing-masing parameter yang mewakili fitur tekstur yaitu *dissimilarity*, *correlation*, *homogeneity*, *contrast*, ASM, dan *energy* masing-masing menggunakan Persamaan (2.37), (2.38), (2.43), (2.44), (2.45), dan (2.46) sehingga diperoleh hasil pada Persamaan (4.13), (4.14), (4.15), (4.16), (4.17), dan (4.18).

$$dissimilarity = \sum_{i=0}^{255}\sum_{j=0}^{255}|i-j|P(i,j) = 12{,}678 \tag{4.13}$$

$$correlation = \frac{\sum_{i=0}^{255}\sum_{j=0}^{255}(i-\mu_x)\left(j-\mu_y\right)P(i,j)}{\sqrt{\delta_x\delta_y}} = 0{,}867 \tag{4.14}$$

$$homogeneity = \sum_{i=0}^{255}\sum_{j=0}^{255}\frac{P(i,j)}{1+(i-j)^2} = 0{,}323 \tag{4.15}$$

$$contrast = \sum_{i=0}^{255}\sum_{j=0}^{255}(i-j)^2\,P(i,j) = 497{,}731 \tag{4.16}$$

$$ASM = \sum_{i=0}^{255}\sum_{j=0}^{255}P(i,j)^2 = 0{,}071 \tag{4.17}$$

$$Energy = \sqrt{0{,}07086} = 0{,}266 \tag{4.18}$$

Langkah-langkah ini selanjutnya dilakukan juga menggunakan sudut 45°, 90°, dan 135°. Dengan demikian, diperoleh hasil ekstraksi fitur tekstur berupa vektor dengan $6 \times 4 = 24$ elemen. Pada citra sampel diperoleh vektor fitur berikut.

$$\begin{bmatrix} 12{,}678 \\ 13{,}909 \\ 13{,}628 \\ 14{,}375 \\ 0{,}867 \\ 0{,}845 \\ 0{,}853 \\ 0{,}836 \\ 0{,}323 \\ 0{,}310 \\ 0{,}318 \\ 0{,}306 \\ 497{,}731 \\ 572{,}652 \\ 548{,}958 \\ 608{,}293 \\ 0{,}071 \\ 0{,}067 \\ 0{,}071 \\ 0{,}065 \\ 0{,}267 \\ 0{,}258 \\ 0{,}267 \\ 0{,}256 \end{bmatrix}$$

.

4.3 Klasifikasi Citra

Proses klasifikasi citra pada penelitian ini dilakukan menggunakan algoritma SVM. Kernel yang dipilih merupakan kernel linear, hal ini dilakukan karena data yang digunakan sudah memiliki dimensi tinggi yaitu 29 dimensi yang berisi tiga fitur warna, dua fitur bentuk, dan 24 fitur tekstur. Selain itu, untuk mengatasi *overfitting*, digunakan *soft margin* seperti tertulis pada Persamaan (2.5). Nilai C yang digunakan pada penelitian ini yaitu 1. Dengan demikian, Persamaan (2.5) dapat ditulis ulang menjadi Persamaan (4.19).

$$Q(w) = \frac{1}{2}\|w\|^2 + \sum_i \xi_i \tag{4.19}$$

Sebelum dilakukan proses klasifikasi, mengingat SVM merupakan model klasifikasi yang hanya bekerja untuk kelas biner, maka data akan dikategorikan ke

dalam 2 kelas. Penelitian ini memiliki data yang terbagi menjadi 5 kelas. Dengan demikian, metode pemisahan kelas terdapat dua cara. Cara pertama yaitu dengan memisahkan salah satu kelas dengan menggabungkan keempat kelas lainnya menjadi satu kelas. Sedangkan cara kedua yaitu menggabungkan dua kelas berdekatan menjadi satu kelas sedangkan tiga kelas lainnya akan digabungkan.

Selanjutnya dilakukan proses latihan terhadap data latih. Untuk itu, pertama-tama nilai *w* dan *b* awal dipilih secara acak. Sebagai contoh, digunakan nilai $b_0 = 1$ dan

$$w_0 = \begin{bmatrix} 1{,}945 \\ 0{,}092 \\ 1{,}659 \\ 1{,}146 \\ 0{,}182 \\ 1{,}409 \\ 0{,}091 \\ 0{,}488 \\ 1{,}928 \\ 0{,}529 \\ 1{,}040 \\ 0{,}572 \\ 0{,}914 \\ 0{,}068 \\ 0{,}057 \\ 0{,}340 \\ 1{,}296 \\ 1{,}143 \\ 0{,}313 \\ 0{,}156 \\ 1{,}547 \\ 0{,}835 \\ 0{,}892 \\ 0{,}681 \\ 0{,}015 \\ 1{,}945 \\ 0{,}095 \\ 0{,}327 \\ 1{,}778 \end{bmatrix}$$

Setelah diperoleh nilai w dan b secara acak, dengan menggunakan Persamaan (2.12) dan (2.13) dengan $\lambda = 1$ dapat diperoleh nilai w dan b selanjutnya seperti tertulis pada Persamaan (4.20) dan (4.21).

$$w_1 = w_0 + 1\left(\|w_0\| + \sum_i \frac{\partial[max(1-y_i({w_0}^T x_i+b),0)]}{\partial w}\right) = \begin{bmatrix} 2{,}524 \\ 0{,}581 \\ 1{,}933 \\ 0{,}006 \\ 0{,}058 \\ 0{,}678 \\ 0{,}580 \\ 0{,}174 \\ 0{,}921 \\ 0{,}652 \\ 0{,}717 \\ 0{,}216 \\ 1{,}347 \\ 0{,}821 \\ 0{,}564 \\ 0{,}853 \\ 0{,}123 \\ 0{,}180 \\ 1{,}217 \\ 0{,}913 \\ 0{,}817 \\ 0{,}438 \\ 0{,}238 \\ 0{,}029 \\ 0{,}368 \\ 1{,}096 \\ 1{,}458 \\ 0{,}422 \\ 1{,}001 \end{bmatrix} \quad (4.20)$$

$$b_1 = b_0 + 1\left(\sum_i \frac{\partial[max(1-y_i(w^T x_i+b),0)]}{\partial b}\right) = 0{,}946 \quad (4.21)$$

dimana

$$\|w_0\| = \sqrt{\begin{matrix} 1{,}945^2 + 0{,}092^2 + 1{,}659^2 + 1{,}146^2 + 0{,}182^2 \\ +1{,}409^2 + 0{,}091^2 + 0{,}488^2 + 1{,}928^2 + 0{,}529^2 \\ +1{,}040^2 + 0{,}572^2 + 0{,}914^2 + 0{,}068^2 + 0{,}057^2 \\ +0{,}340^2 + 1{,}296^2 + 1{,}143^2 + 0{,}313^2 + 0{,}156^2 \\ +1{,}547^2 + 0{,}835^2 + 0{,}892^2 + 0{,}681^2 + 0{,}015^2 \\ +1{,}945^2 + 0{,}095^2 + 0{,}327^2 + 1{,}778^2 \end{matrix}} = 31{,}005311$$

Proses tersebut dilakukan berulang sehingga diperoleh model SVM yang memisahkan dua kelas dengan nilai w dan b yang optimal. Selanjutnya hal ini

dilakukan berulang kali terhadap setiap 2 kelas yang telah terbagi sehingga dihasilkan 10 model SVM.

Setelah proses latihan selesai, selanjutnya model SVM akan diterapkan terhadap data uji. Setiap data uji akan diklasifikasikan menggunakan masing-masing 10 model SVM. Hasil akhir dari proses klasifikasi dapat berupa vektor berukuran 10×1 yang berisikan hasil klasifikasi terhadap masing-masing model SVM. Vektor hasil klasifikasi ini berisi nilai antara -1 dan 1. Dari hasil tersebut dapat ditentukan data tersebut terklasifikasi pada kelas kencur, temulawak, kunyit, lengkuas, atau jahe. Sebagai contoh visualisasi klasifikasi berdasarkan hasil akhir vektor dapat dilihat pada Gambar 4.9.

[1,-1,-1,-1,-1,1,1,-1,-1,-1] = Kencur
[-1,1,-1,-1,-1,-1,1,1,-1,-1] = Temulawak
[-1,-1,1,-1,-1,-1,-1,1,1,-1] = Kunyit
[-1,-1,-1,1,-1,-1,-1,-1,1,1] = Lengkuas
[-1,-1,-1,-1,1,1,-1,-1,-1,1] = Jahe

Gambar 4.9 Contoh Pembagian Kelas Hasil Klasifikasi

Pada penelitian ini, masing-masing label rimpang diubah menjadi angka $0 - 4$ dimana label 0 merepresentasikan jahe, label 1 merepresentasikan kencur, label 2 merepresentasikan kunyit, label 3 merepresentasikan lengkuas dan label 4 merepresentasikan temulawak. Dengan demikian, hasil proses klasifikasi terhadap 25 data uji dapat direpresentasikan sebagaimana Tabel 4.1 (selengkapnya dapat dilihat pada Lampiran 6).

Tabel 4.1 Tabel Hasil Proses Klasifikasi Jenis Tanaman Rimpang

Data Sampel	Label	Prediksi
	0	0
	0	0
	1	1
	1	1
	2	2
	2	2
	3	3

Lanjutan Tabel 4.1 Tabel Hasil Proses Klasifikasi Jenis Tanaman Rimpang

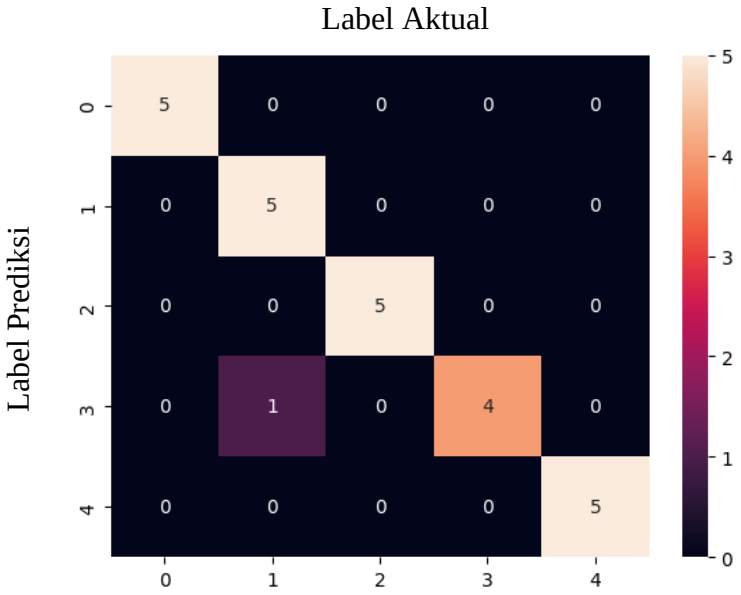
	3	1
	4	4
	4	4

4.4 Evaluasi

Salah satu metode evaluasi dapat dilakukan berdasarkan nilai akurasi yang telah didefinisikan pada Persamaan (2.14). Pada penelitian ini proses evaluasi dibantu dengan algoritma *K-fold Cross Validation*. Tujuan algoritma ini adalah untuk mendapatkan nilai akurasi yang akurat berdasarkan rata-rata akurasi yang diperoleh dari k kali klasifikasi. Nilai k yang dipilih yaitu 5 sehingga data akan dibagi menjadi lima bagian dimana setiap bagian akan mendapatkan peluang yang sama untuk digunakan sebagai data latih maupun data uji. Dengan menggunakan $k = 5$ maka pada setiap proses klasifikasi, data akan dibagi menjadi 80% data latih dan 20% data uji. Dengan demikian, pada setiap proses klasifikasi akan digunakan 100 data latih dan 25 data uji.

Pada penelitian ini, proses latihan dan pengujian pertama menghasilkan *confusion matrix* pada Gambar 4.10. Label 0 pada sumbu x dan sumbu y

merepresentasikan label jahe, label 1 merepresentasikan kencur, label 2 mewakili kunyit, label 3 mewakili lengkuas dan label 4 merepresentasikan temulawak.



Gambar 4.10 Hasil *Confusion Matrix*

Berdasarkan *confusion matrix* tersebut, setiap elemen baris mewakili hasil prediksi/klasifikasi sedangkan setiap elemen kolomnya mewakili label aslinya. Setiap data yang terklasifikasi dengan benar akan tertulis pada *confusion matrix* dengan koordinat baris dan kolom yang sama. Dengan demikian, pada *confusion matrix* tersebut memberitahukan bahwasanya terdapat lima klasifikasi benar untuk label 0 yaitu jahe, lima klasifikasi benar untuk label 1 yaitu kencur, lima klasifikasi benar untuk label 2 yaitu kunyit, empat klasifikasi benar untuk label 3 yaitu lengkuas, dan lima klasifikasi benar untuk label 4 yaitu temulawak. Maka, berdasarkan *confusion matrix* tersebut diperoleh 24 data terklasifikasi benar dan 1 data terklasifikasi salah, sehingga klasifikasi tersebut menghasilkan nilai akurasi seperti tertulis pada Persamaan (4.22).

$$akurasi_1 = \frac{24}{25} = 0{,}96 \tag{4.22}$$

Langkah evaluasi diterapkan terhadap empat proses *cross validation* lainnya. Pada penelitian ini, dihasilkan nilai akurasi masing-masing proses klasifikasi sebagai berikut.

$$akurasi_1 = 0{,}96$$

$$akurasi_2 = 0{,}96$$

$$akurasi_3 = 1{,}00$$

$$akurasi_4 = 0{,}96$$

$$akurasi_5 = 0{,}96$$

Dengan demikian, diperoleh hasil akurasi rata-rata proses klasifikasi jenis tanaman rimpang berdasarkan fitur warna, bentuk, dan tekstur dengan menggunakan model SVM adalah $\frac{0{,}96+0{,}96+1{,}00+0{,}96+0{,}96}{5} = 0{,}968$ atau juga dapat ditulis dengan 96,8%.

4.5 Implementasi *Support Vector Machine* dalam Pandangan Islam

Proses implementasi SVM pada klasifikasi jenis tanaman rimpang berdasarkan fitur warna, bentuk dan tekstur dapat membuka wawasan baru dalam bidang ilmu pengetahuan. Hal ini selaras dengan salah satu perintah Allah SWT dalam agama Islam yaitu menuntut ilmu. Dalam agama Islam, menuntut ilmu merupakan suatu kewajiban bagi setiap muslim. Mengembangkan ilmu merupakan salah satu cara bagi manusia untuk bersyukur atas segala ciptaan Allah SWT dan mengagungkan kebesaran Allah SWT. Hal ini bersesuaian dengan Al-Qur'an surat Ali Imran ayat 190 yang berbunyi (Kemenag RI, 2022):

اِنَّ فِيْ خَلْقِ السَّمٰوٰتِ وَالْاَرْضِ وَاخْتِلَافِ الَّيْلِ وَالنَّهَارِ لَاٰيٰتٍ لِّاُولِى الْاَلْبَابِۙ

"Sesungguhnya dalam penciptaan langit dan bumi serta pergantian malam dan siang terdapat tanda-tanda (kebesaran Allah) bagi orang yang berakal," (Q.S. Ali Imran:190)

Selain itu, implementasi SVM pada klasifikasi jenis tanaman rimpang berdasarkan fitur warna, bentuk dan tekstur juga dapat menciptakan teknologi yang dapat memudahkan urusan manusia khususnya dalam membedakan jenis tanaman rimpang. Dengan demikian, penelitian ini memberikan manfaat dan bantuan secara tidak langsung terhadap para pembacanya. Sebagaimana hadits Rasulullah SAW. yang menjelaskan akan keutamaan saling tolong menolong sesama manusia berikut (An-Nawawi, 1999).

عَنْ أَبِي هُرَيْرَةَ رَضِيَ اللهُ عَنْهُ، عَنِ النَّبِيِّ ﷺ قَالَ: «مَنْ نَفَّسَ عَنْ مُؤْمِنٍ كُرْبَةً مِنْ كُرَبِ الدُّنْيَا، نَفَّسَ اللهُ عَنهُ كُرْبَةً مِنْ كُرَبِ يَوْمِ القِيَامَةِ. وَمَنْ يَسَّرَ عَلَى مُعْسِرٍ، يَسَّرَ اللهُ عَلَيْهِ فِي الدُّنْيَا وَالآخِرَةِ. وَمَنْ سَتَرَ مُسْلِماً سَتَرَهُ اللهُ فِي الدُّنْيَا وَالآخِرَةِ. وَاللهُ فِي عَوْنِ العَبْدِ مَا كَانَ العَبْدُ فِي عَوْنِ أَخِيْهِ. وَمَنْ سَلَكَ طَرِيْقاً يَلْتَمِسُ فِيْهِ عِلْماً سَهَّلَ اللهُ لَهُ بِهِ طَرِيْقاً إِلَى الجَنَّةِ. وَمَا اجْتَمَعَ قَوْمٌ فِي بَيْتٍ مِنْ بُيُوْتِ اللهِ يَتْلُوْنَ كِتَابَ اللهِ وَيَتَدَارَسُوْنَهُ بَيْنَهُمْ إِلَّا نَزَلَتْ عَلَيْهِمُ السَّكِيْنَةُ، وَغَشِيَتْهُمُ الرَّحْمَةُ، وَحَفَّتْهُمُ الْمَلَائِكَةُ، وَذَكَرَهُمُ اللهُ فِيْمَنْ عِنْدَهُ، وَمَنْ بَطَّأَ بِهِ عَمَلُهُ لَمْ يُسْرِعْ بِهِ نَسَبُهُ» رَوَاهُ مُسْلِمٌ بِهَذَا اللَّفْظِ

"Dari Abu Hurairah radhiyallahu 'anhu, dari Nabi shallallahu 'alaihi wa sallam bersabda, "Barangsiapa yang menghilangkan kesusahan dari kesusahan-kesusahan dunia orang mukmin, maka Allah akan menghilangkan kesusahan dari kesusahan-kesusahan hari kiamat. Barangsiapa yang memberi kemudahan orang yang kesulitan (utang), maka Allah akan memberi kemudahan baginya di dunia dan akhirat. Siapa yang menutup aib seorang muslim, maka Allah akan menutup aibnya di dunia dan di akhirat. Siapa saja yang menolong saudaranya, maka Allah akan menolongnya sebagaimana ia menolong saudaraya. Barangsiapa yang menempuh perjalanan dalam rangka menuntut ilmu, maka Allah akan mudahkan baginya jalan menuju surga. Tidaklah berkumpul sekelompok orang di salah satu rumah Allah (masjid) untuk membaca Kitabullah dan saling mempelajarinya di antara mereka, melainkan akan turun kepada mereka ketenangan, rahmat meliputinya, para malaikat mengelilinginya, dan Allah menyanjung namanya kepada Malaikat yang ada di sisi-Nya. Barangsiapa yang lambat amalnya, maka tidak akan bisa dikejar oleh nasabnya (garis keturunannya yang mulia)." (H.R. Muslim)

Berdasarkan hadits tersebut, implementasi SVM merupakan salah satu hal yang dianjurkan dalam pandangan Islam. Hal ini bersesuaian dengan prinsip belajar, mengembangkan ilmu, dan tolong menolong sesama manusia. Dengan demikian, dengan adanya penelitian ini diharapkan teknologi dan ilmu pengetahuan akan dapat berkembang semakin pesat sehingga dapat memberi manfaat lebih banyak terhadap umat manusia.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan seluruh penjelasan pada bab 4, diperoleh kesimpulan bahwa proses implementasi algoritma *support vector machine* pada klasifikasi jenis tanaman rimpang berdasarkan fitur warna, bentuk, dan tekstur dapat menghasilkan nilai akurasi sebesar 96,8%. Hal ini dapat diperoleh dengan menggunakan vektor fitur masing-masing data yang berukuran 29×1 yang terdiri dari tiga fitur warna, dua fitur bentuk, dan 24 fitur tekstur. Masing-masing fitur memiliki peran penting untuk dapat mencapai akurasi di atas 90%. Semakin akurat dan semakin detil/banyak fitur yang digunakan, maka nilai akurasi dapat menjadi semakin baik pula. Selain itu, algoritma SVM juga menyediakan kemampuan dalam memisahkan kelas data dengan sangat baik. Meskipun SVM memiliki kekurangan dalam hal efektivitas waktu untuk menghasilkan sebanyak 10 model, setiap model SVM membantu proses klasifikasi menjadi semakin akurat. Dengan demikian, penelitian ini berhasil mendapatkan akurasi lebih tinggi daripada penelitian sebelumnya yang dilakukan menggunakan algoritma *Random Forest* dan hanya dilakukan berdasarkan fitur warna dan tekstur saja.

5.2 Saran

Pada penelitian ini, waktu *training* secara komputasi berjalan cepat yaitu $\pm 7{,}6$ milidetik dan waktu *testing* yaitu $\pm 0{,}7$ milidetik. Sedangkan *preprocessing* dan ekstrasi fitur memerlukan waktu sebanyak 2 menit 58 detik dengan banyak data

125 citra dimana rata-rata proses *preprocessing* dan ekstraksi fitur setiap citra adalah 1,424 detik pada setiap citra. Hal ini dikarenakan banyaknya tahapan yang perlu dilakukan dalam proses ini. Hal ini akan berdampak besar jika diterapkan dalam dunia industri dengan data yang berjumlah besar. Oleh karena itu, saran peneliti terhadap penelitian selanjutnya yaitu menggunakan proses ekstraksi fitur yang memiliki efisiensi waktu yang lebih efisien. Selain itu, saran selanjutnya adalah menggunakan fitur lain dari rimpang selain fitur warna, bentuk, maupun tekstur.

DAFTAR PUSTAKA

Abe, Shigeo. (2005). *Support Vector Machines for Pattern Classification*. London: Springer

Andono, N.P., T.Sutojo dan Muljono. (2017). *Pengolahan Citra Digital*. Yogyakarta: Andi

An-Nawawi, Al-Imam Abu Zakaria Yahya bin Syaraf. (1999). *Riyadhus Sholihin*. Jakarta: Pustaka Amani

Arsy, L., Nurhayati, O.D., dan Martono, K.T. (2016). *Aplikasi Pengolahan Citra Digital Meat Detection Dengan Metode Segmentasi K-Mean Clustering Berbasis OpenCV Dan Eclipse*. Jurnal Teknologi dan Sistem Komputer, 4(2), 322-332. https://doi.org/10.14710/jtsiskom.4.2.2016.322-332

Chernov, V., Alander, J., & Bochko, V. (2015). *Integer-based accurate conversion between RGB and HSV color spaces*. Computers & Electrical Engineering, 46, 328–337. doi:10.1016/j.compeleceng.2015.08.005

García-Gonzalo, Esperanza, dkk. (2016). *Hard-Rock Stability Analysis for Span Design in Entry-Type Excavations with Learning Classifiers.* Materials. 9. 531. 10.3390/ma9070531.

Hasty. (2023). *Accuracy | Hasty.ai Documentation*. Diambil dari https://hasty.ai/docs/mp-wiki/metrics/accuracy

Hikmatulloh, E., Lasmanawati, E., dan Setiawati, T. (2017). *Manfaat Pengetahuan Bumbu dan Rempah Pada Pengolahan Makanan Indonesia Siswa SMKN 9 Bandung*. Media Pendidikan, Gizi dan Kuliner Vol. 6, No. 1: Hal. 42-50

Ichwan, M., Dewi, I.A., dan Muharom S, Zeni. (2018). *Klasifikasi Support Vector Machine (SVM) untuk Menentukan Tingkat Kemanisan Mangga berdasarkan Fitur Warna*. MIND Journal Vol. 3 No. 2: Hal. 16-24

Kemenag RI. (2022). *Qur'an Kemenag*. Lajnah Pentashihan Mushaf Al-Qur'an. Diambil dari https://quran.kemenag.go.id/

Kaharuddin, Kusrini, dan Luthfi, E.T. (2019). *Klasifikasi Jenis Rempah-rempah berdasarkan Fitur Warna RGB dan Tekstur menggunakan Algoritma K-Nearest Neighbor.* Jurnal Informasi Interaktif Vol. 4, No. 1: Hal. 17-22

Kaharuddin, K., Wati, V., Pawan, E., dan Hasan, P. (2019) *Classification of Spice Types using K-Nearest Neighbor Algorithm*. In 2019 International Conference on Information and Communications Technology, ICOIACT 2019 (pp. 285–290). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICOIACT46704.2019.8938515

Kundu, Rohit. (2022). *Confusion Matrix: How To Use It & Interpret Results [Examples]*. Diambil dari https://www.v7labs.com/blog/confusion-matrix-guide

Kusuma, Purba Daru. (2020). *Machine Learning Teori, Program, dan Studi Kasus*. Yogyakarta: Deepublish

Mathmonks. (2023). *Ellipse - Definition, Parts, Equations, and Diagrams*. Diambil dari https://mathmonks.com/ellipse#google_vignette

Misra, Rishabh. (2019). *Support Vector Machines - Soft Margin Formulation and Kernel Trick*. Diambil dari https://towardsdatascience.com/support-vector-machines-soft-margin-formulation-and-kernel-trick-4c9729dc8efe

Murphy, Kevin P. (2012). *Machine Learning: A Probabilistic Perspective*. London: The MIT Press

Pamungkas, Adi. (2017). *Segmentasi Citra dengan Metode Thresholding*. Diambil dari https://pemrogramanmatlab.com/2017/04/08/segmentasi-citra-dengan-metode-thresholding/

Pandian, Shantababu. (2022). *K-Fold Cross Validation Technique and its Essentials*. Diambil dari https://www.analyticsvidhya.com/blog/2022/02/k-fold-cross-validation-technique-and-its-essentials/#h-what-is-k-fold-cross-validation

Rachmawanto, E.H., Anarqi, G.R., Setiadi, D.R.I.M., dan Sari, C.A. (2018). *Handwriting Recognition Using Eccentricity and Metric Feature Extraction Based on K-Nearest Neighbors*. 2018 International Seminar on Application for Technology of Information and Communication. doi:10.1109/isemantic.2018.8549804

Ranjan, Gopal Krishna. (2021). *Introduction to K-fold Cross Validation in Python*. Diambil dari https://sqlrelease.com/introduction-to-k-fold-cross-validation-in-python#google_vignette

Rizki, A. (2022, Mei 25). *Rizhoma: Pengertian, Ciri-ciri, dan Contohnya*. Retrieved from detik.com: https://www.detik.com/edu/detikpedia/d-6094802/rhizoma-pengertian-ciri-ciri-dan-contohnya

Setiawan, Rudi. (2020). *Rempah Indonesia di Pasar Dunia: Memudahkan dalam Pencarian Pasar Rempah-Rempah*. Jakarta: Tanamiku Books

Softscients. (2021). *Ciri Fitur dengan GLCM - Gray Level Co-occurrence Matrix*. Diambil dari https://softscients.com/2021/09/15/ciri-fitur-dengan-glcm-gray-level-coocurance-matrix/

Verma, M., Raman, B., dan Murala, S. (2015). *Local Extrema Co-occurrence*

Pattern for Color and Texture Image Retrieval. Neurocomputing. 165. 255–269. 10.1016/j.neucom.2015.03.015.

Yousef, Jamileh. (2011). *Image Binarization using Otsu Thresholding Algorithm*. Canada: University of Guelph

Yunus, Muhammad. (2020). *Feature Extraction: Gray Level Co-occurrence Matrix*. Diambil dari https://yunusmuhammad007.medium.com/feature-extraction-gray-level-co-occurrence-matrix-glcm-10c45b6d46a1

LAMPIRAN

Lampiran 1 Matriks Citra Sampel

https://bit.ly/matriks_citra_sampel

Lampiran 2 Matriks Citra Sampel berformat *Grayscale*

https://bit.ly/matriks_citra_sampel_grayscale

Lampiran 3 Matriks GLCM Non Simetris

https://bit.ly/glcm_nonsymmetric

Lampiran 4 Matriks GLCM Simetris

https://bit.ly/glcm_symmetric

Lampiran 5 Matriks GLCM Ternormalisasi

https://bit.ly/glcm_normed

Lampiran 6 Hasil Klasifikasi

https://bit.ly/hasil_klasifikasi_rimpang

Lampiran 7 Kode *Python*

Impor *Library*

```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import os
import fnmatch
import glob
from tqdm import tqdm
from google.colab.patches import cv2_imshow
import seaborn as sns
from skimage.feature import graycomatrix, graycoprops
from skimage.measure import label, regionprops, regionprops_table
from sklearn import preprocessing
from sklearn import svm
```

```
from sklearn import metrics
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import cross_val_score
```

Fungsi Segmentasi Citra

```
def segmentation(img, k):
  hsv = cv.cvtColor(img, cv.COLOR_BGR2HSV)
  pixel_vals = hsv.reshape((-1,3))
  pixel_vals = np.float32(pixel_vals)
  criteria = (cv.TERM_CRITERIA_EPS + cv.TERM_CRITERIA_MAX_ITER,
100, 0.99)
  ret, label, center = cv.kmeans(pixel_vals, k, None, criteria,
10, cv.KMEANS_RANDOM_CENTERS)
  center = np.uint8(center)
  segmented_data = center[label.flatten()]
  segmented_image = segmented_data.reshape((img.shape))
  otsu, binary_image =
cv.threshold(cv.cvtColor(segmented_image,cv.COLOR_BGR2GRAY),0,25
5,cv.THRESH_OTSU) # melakukan thresholding
  kernel = np.ones((15,15),np.uint8)
  mask = cv.morphologyEx(binary_image,cv.MORPH_CLOSE,kernel)
  dst = cv.bitwise_and(img,img,mask=mask) # melakukan masking
  contours, hierarchy =
cv.findContours(mask,cv.RETR_TREE,cv.CHAIN_APPROX_SIMPLE)
  selected = max(contours,key=cv.contourArea)
  x,y,w,h = cv.boundingRect(selected)
  mask = mask[y:y+h,x:x+w]
  result = dst[y:y+h,x:x+w]
  return (mask, result)
```

Fungsi Ekstraksi Fitur Warna

```
def extract_color(img):
  blue, green, red = cv.split(img)
  area = red != 0
  result = [np.mean(red[area]), np.mean(green[area]),
np.mean(blue[area])]
  return result
```

Fungsi Ektraksi Fitur Bentuk

```
def extract_shape(mask):
  label_img = label(mask)
  props = regionprops(label_img)
  eccentricity = getattr(props[0],'eccentricity')
  area = getattr(props[0],'area')
```

```
  perimeter = getattr(props[0],'perimeter')
  metric = (4*np.pi*area)/(perimeter**2)
  result = [eccentricity, metric]
  return result
```

Fungsi Ekstraksi Fitur Tekstur

```
def extract_texture(img):
  gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
  glcm_properties = [
      'dissimilarity',
      'correlation',
      'homogeneity',
      'contrast',
      'ASM',
      'energy'
  ]
  glcm = graycomatrix(gray,
                      distances=[5],
                      angles=[0,np.pi/4,np.pi/2,3*np.pi/4],
                      levels=256,
                      symmetric=True,
                      normed=True)
  result = [propery for name in glcm_properties for propery in
graycoprops(glcm, name)[0]]
  return result
```

Fungsi Ekstraksi Fitur

```
def extract_feature(img, mask):
  feature1 = extract_color(img)
  feature2 = extract_texture(img)
  feature3 = extract_shape(mask)
  result = np.array(feature1 + feature2 + feature3)
  return result
```

Proses Impor Data

```
dir_train = os.path.join(dataset, 'train')
dir_test = os.path.join(dataset, 'test')

# Capture training, testing data and their labels into respective
lists
train_images = []
test_images = []
train_labels = []
test_labels = []
```

```
print("------------------------------------------------------------
--------")
print("Importing training images...\n")

for dir in tqdm(os.listdir(os.path.join(dir_train))):
  dir_path = os.path.join(dir_train, dir)
  for img_path in glob.glob(os.path.join(dir_path, "*.jpg")):
    img = cv.imread(img_path)
    train_images.append(img)
    train_labels.append(dir)

print("------------------------------------------------------------
--------")
print("Importing testing images...\n")

for dir in tqdm(os.listdir(os.path.join(dir_test))):
  dir_path = os.path.join(dir_test, dir)
  for img_path in glob.glob(os.path.join(dir_path, "*.jpg")):
    img = cv.imread(img_path)
    test_images.append(img)
    test_labels.append(dir)

# Convert lists to arrays
train_images = np.array(train_images)
test_images = np.array(test_images)
train_labels = np.array(train_labels)
test_labels = np.array(test_labels)

print("\n------------------------------------------------------------
----------")
print("Importing DONE!")
```

Proses Ekstraksi Fitur

```
x_train = []
x_test = []

k = 5 # banyak segmentasi

print("------------------------------------------------------------
--------")
print("Extracting training images features...\n")

for img in tqdm(train_images):
  mask, segmented = segmentation(img, k)
  feature = extract_feature(segmented, mask)
  x_train.append(feature)
```

```
print("-------------------------------------------------------------
--------")
print("Extracting testing images features...\n")

for img in tqdm(test_images):
  mask, segmented = segmentation(img, k)
  feature = extract_feature(segmented, mask)
  x_test.append(feature)

print("\n-------------------------------------------------------------
----------")
print("Extracting features DONE!")

#Encode labels from text to integers.

le = preprocessing.LabelEncoder()
le.fit(train_labels)
train_labels_encoded = le.transform(train_labels)
le.fit(test_labels)
test_labels_encoded = le.transform(test_labels)

#Split data into test and train datasets (already split but
assigning to meaningful convention)

x_train, y_train, x_test, y_test = np.array(x_train),
train_labels_encoded, np.array(x_test), test_labels_encoded
```

Proses Klasifikasi menggunakan SVM

```
model = svm.SVC(kernel='linear', C=1) # membuat model
model.fit(x_train, y_train) # proses training
prediction = model.predict(x_test) # proses testing
y_pred = le.inverse_transform(prediction) # hasil prediksi
```

Proses Evaluasi Percobaan Awal

```
#Menghitung akurasi
print ("Accuracy = ", metrics.accuracy_score(y_test, prediction)
* 100)
#Confusion Matrix - verify accuracy of each class
cm = confusion_matrix(test_labels, y_pred)
sns.heatmap(cm, annot=True)
```

Proses Evaluasi dengan K-*fold Cross Validation*

```
X = np.concatenate((x_train, x_test))
y = np.concatenate((y_train, y_test))
```

```
lin_svm = clf = svm.SVC(kernel='linear', C=1)
scores = cross_val_score(lin_svm, X, y, cv=5)
final_scores = np.mean(scores)
```

RIWAYAT HIDUP

Sania Kamelia Hamdani atau Nia, lahir di Sleman pada tanggal 18 Januari 2002, bertempat tinggal di Jl. Burung Greja No. 1, Kelurahan Arjowinangun, Kecamatan Kedungkandang, Kota Malang. Penulis merupakan anak kedua dari pasangan Bapak Uuj Hamdanie dan Ibu Maryani. Penulis telah menempuh pendidikan dasar di SDN Model Kota Malang, kemudian melanjutkan menengah pertama dan menengah atas berturut-turut di SMP dan SMA Al Munawwariyyah, lulus pada tahun 2019. Setelah itu, penulis melanjutkan pendidikan sebagai mahasiswa program studi Matematika di Universitas Islam Negeri Maulana Malik Ibrahim Malang. Selama menempuh pendidikan Strata 1, penulis aktif mengikuti kegiatan-kegiatan keorganisasian di dalam maupun di luar kampus. Beberapa pengalaman keorganisasian yang pernah diikuti oleh penulis antara lain yaitu sebagai pengurus Himpunan Mahasiswa Jurusan (HMJ) "Integral" Matematika, pengurus Dewan Eksekutif Mahasiswa Fakultas Sains dan Teknologi (DEMA-FST), pengurus Komunitas Literasi Matematika, ketua Komunitas *Mathematic English Club* (MEC) serta relawan Senyum Anak Nusantara *Chapter* Malang. Selain itu, peneliti juga memiliki pengalaman sebagai Asisten Laboratorium pada mata kuliah Pemrograman Komputer.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

BUKTI KONSULTASI SKRIPSI

Nama	: Sania Kamelia Hamdani
NIM	: 19610071
Fakultas / Program Studi	: Sains dan Teknologi / Matematika
Judul Skripsi	: Implementasi Algoritma *Support Vector Machine* pada Klasifikasi Jenis Tanaman Rimpang berdasarkan Fitur Warna, Bentuk, dan Tekstur
Pembimbing I	: Hisyam Fahmi, M.Kom.
Pembimbing II	: M. Nafie Jauhari, M.Si.

No	Tanggal	Hal	Tanda Tangan
1.	3 November 2022	ACC Pengajuan Topik	1.
2.	14 April 2023	Konsultasi Bab I, II, dan III	2.
3.	9 Mei 2023	Revisi Bab I, II, dan III	3.
4.	10 Mei 2023	ACC Bab I, II, dan III	4.
5.	25 Mei 2023	ACC Kajian Agama Bab I dan II	5.
6.	18 Agustus 2023	ACC Revisi Seminar Proposal	6.
7.	30 Agustus 2023	Konsultasi Bab IV dan V	7.
8.	12 September 2023	Revisi Bab IV dan V	8.
9.	26 September 2023	ACC Bab IV dan V	9.
10.	26 September 2023	Konsultasi Kajian Agama Bab IV	10.
11.	11 Oktober 2023	ACC Kajian Agama Bab IV	11.
12.	27 November 2023	Revisi Seminar Hasil	12.
13.	1 Desember 2023	ACC Revisi Seminar Hasil	13.
14.	27 Desember 2023	ACC Keseluruhan (Pembimbing I)	14.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

15.	27 Desember 2023	ACC Keseluruhan (Pembimbing II)	15.

Malang, 27 Desember 2023
Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc
NIP. 19741129 200012 2 005