AnoaTIK: Jurnal Teknologi Informasi dan Komputer e-ISSN: **2987-7652**
Vol. 2, No. 2, Desember 2024 Hal. 101-110

KLASIFIKASI TINGKAT KEMATANGAN BUAH NAGA BERDASARKAN WARNA KULIT MENGGUNAKAN ALGORITMA *K-NEAREST NEIGHBORS*

Erick Hernando[1] , Ade Chandra Saputra[2], Jadiaman Parhusip [3]

[1,2,3] Jurusan Teknik Informatika, Fakultas Teknik, Universitas Palangka Raya
Kampus Tunjung Nyaho Jl. Yos Sudarso, Palangka Raya 73112
Email: [1] erickhernando1@mhs.eng.upr.ac.id, [2] adechandra@it.upr.ac.id, [3] parhusip.jadiaman@it.upr.ac.id
** Penulis Korespondensi*

Abstrak

Penelitian ini bertujuan untuk mengembangkan sebuah metode yang efektif dalam menentukan tingkat kematangan buah naga pada kelas layak panen, matang, mentah secara otomatis dengan memanfaatkan algoritma *K-Nearest Neighbors* (K-NN) melalui proses *Knowledge Discovery in Databases* (KDD). Proses KDD, yang melibatkan serangkaian langkah mulai dari pemilihan data, praproses data, transformasi data, hingga penerapan algoritma untuk menghasilkan pengetahuan yang bermanfaat, digunakan dalam penelitian ini untuk mengolah dan menganalisis data citra buah naga. Pada penelitian ini klasifikasi diolah melalui tahapan KDD, termasuk proses praproses untuk membersihkan dan mempersiapkan data, Penggunaan *Min-Max Normalization* untuk menstandarisasi data sehingga semua fitur berada pada skala yang sama, sangat penting untuk kinerja model K-NN, transformasi untuk mengekstraksi data kelas, dan penerapan algoritma K-NN untuk klasifikasi kematangan buah. Pemilihan algoritma K-NN dalam tahapan KDD didasarkan pada kesederhanaan serta kemampuannya dalam mengklasifikasikan data dengan tingkat akurasi yang tinggi. Hasil penelitian menunjukkan bahwa metode KDD yang diterapkan dengan algoritma K-NN mampu mengklasifikasikan kematangan buah naga dengan Akurasi terbaik diperoleh pada nilai K = 3 dengan persentase akurasi sebesar 91% tanpa memerlukan pemotongan fisik pada buah. Dengan demikian, penelitian ini tidak hanya memberikan kontribusi dalam bidang pertanian presisi tetapi juga menunjukkan bagaimana metode KDD dapat diterapkan secara efektif untuk menyelesaikan masalah nyata di lapangan.

Kata Kunci : *K-Nearest Neighbors* (K-NN), *Knowledge Discovery in Databases* (KDD), Klasifikasi kematangan buah naga, Praproses data, Transformasi data, Model warna RGB

Abstract

This research aims to develop an effective method for determining the maturity level of dragon fruit in the harvestable, ripe, raw classes automatically by utilizing the K-Nearest Neighbors (K-NN) algorithm through the Knowledge Discovery in Databases (KDD) process. The KDD process, which involves a series of steps starting from data selection, data preprocessing, data transformation, to applying algorithms to produce useful knowledge, is used in this research to process and analyze dragon fruit image data. In this research the classification is processed through the KDD stages, including a preprocessing process to clean and prepare the data, the use of Min-Max Normalization to standardize the data so that all features are on the same scale, very important for the performance of the K-NN model, transformation to extract class data, and application of the K-NN algorithm for fruit maturity classification. The selection of the K-NN algorithm in the KDD stage is based on its simplicity and ability to classify data with a high level of accuracy. The research results show that the KDD method applied with the K-NN algorithm is able to classify the ripeness of dragon fruit with the best accuracy obtained at a value of K = 3 with an accuracy percentage of 91% without requiring physical cutting of the fruit. Thus, this research not only contributes to the field of precision agriculture but also shows how the KDD method can be applied effectively to solve real problems in the field.

Keywords: *K-Nearest Neighbors (K-NN), Knowledge Discovery in Databases (KDD), Dragon Fruit Ripeness Classification, Data Preprocessing, Data Transformation, RGB Color Model.*

1. PENDAHULUAN

Buah Naga termasuk Salah satu jenis buah yang disukai oleh masyarakat dan juga mempunyai banyak manfaat bagi Kesehatan . Buah ini berbentuk bulat memanjang serta kulitnya agak tebal . Selama ini petani kebun buah naga dalam melakukan pemilihan buah naga yang telah matang pada musim panen terkadang masih memiliki kendala seperti melakukan penyortiran untuk mengidentifikasi mana yang sudah matang atau belum, hal ini dikarenakan pada buah naga terdapat kulit atau teksturnya yang tebal, sehingga tidak efektif dan efisien dalam mengetahui kematangan buah naga dan terlebih lagi mesti harus dibelah buahnya untuk mengetahui sudah matang buah naga tersebut atau belum.

Dalam dunia komputasi modern, algoritma memiliki peran penting dalam berbagai aplikasi, mulai dari pengolahan data, machine learning, hingga artificial intelligence. Pemilihan algoritma yang tepat dapat mempengaruhi kinerja dan efisiensi dari suatu sistem atau aplikasi. Oleh karena itu, penting untuk melakukan perbandingan antara berbagai algoritma untuk menentukan mana yang paling cocok untuk kebutuhan tertentu.

Salah satu bidang yang sering memanfaatkan berbagai algoritma adalah pengenalan pola (pattern recognition) dan klasifikasi data. Di dalamnya, berbagai algoritma digunakan untuk menganalisis dan mengklasifikasikan data berdasarkan fitur-fitur tertentu. Algoritma seperti *K-Nearest Neighbors* (K-NN) adalah beberapa yang sering digunakan dalam bidang ini. Algoritma *K-Nearest Neighbors* (K-NN) adalah salah satu algoritma yang paling sederhana dan intuitif dalam klasifikasi. Algoritma ini bekerja dengan cara mencari 'k' tetangga terdekat dari data yang akan diklasifikasikan dan menentukan kelas berdasarkan mayoritas kelas dari tetangga tersebut. Meskipun sederhana, K-NN seringkali menunjukkan kinerja yang baik pada berbagai jenis data [1].

Banyak metode dalam citra digital yang digunakan dalam penentuan model warna seperti model warna RGB (*Red Green Blue*). Pengolahan warna RGB (*Red Green Blue*) mudah dan juga sederhana, hal itu dapat dilakukan dengan pembacaan pada nilai R (*Red*), G (*Green*), dan B (*Blue*) dalam sebuah pixel. Dalam menampilkan dan mengartikan warna hasil perhitungan sehingga mempunyai arti sesuai yang diinginkan. Sementara dalam pengklasifikasian bertujuan untuk menentukan kelas-kelas yang telah ditentukan dalam tiap-tiap contoh data. Dengan demikian bisa membantu untuk memahami data yang ada dan bisa dipakai dalam memprediksi bagaimana masalah baru akan berperilaku. Dengan adanya pengklasifikasian menggunakan pengolahan citra RGB pada komputer akan mempermudah dalam pemilahan penggolongan buah buah naga yang masih mentah dan sudah matang.

Fitur warna telah banyak diterapkan untuk evaluasi tingkat kematangan buah. Dalam hal ini, fitur kedalaman warna *red*, *green*, dan *blue* (RGB) dapat digunakan untuk mengklasifikasi tingkat kematangan pada buah markisa (Tu et al., 2018). Selain itu, identifikasi ciri berdasarkan fitur warna pada buah naga pernah dilakukan menggunakan algoritma *Multi-Class Support Vector Machine* (SVM) untuk mengklasifikasikan kematangan buah naga berdasarkan warna.

Dalam penelitian ini, penulis akan menggunakan algoritma *K-Nearest Neighbors* (K-NN) dalam klasifikasi data tingkat kematangan buah naga. Klasifikasi ini akan mencakup aspek-aspek seperti akurasi, kecepatan pemrosesan, dan kemudahan implementasi pada Algoritma *K-Nearest Neighbors*.

2. TINJAUAN PUSTAKA

2.1. Buah Naga

Buah naga dengan nama latin *Hylocerus undatus* memiliki bentuk bulat memanjang dengan jumbai atau sisik berwarna hijau pada permukaan kulit. Menurut Martasuta (2000) spesies buah naga yang biasa dikonsumsi berasal dari genus *Hylocereus*, *Selenicereus* dan *Mediocactus*. Kelompok *Hylocereus* memiliki 17 spesies. Tiga di antaranya yang sudah banyak dibudidayakan secara komersial adalah Hylocereus undatus (daging buah putih), Hylocereus polyrhizus (daging buah merah keunguan), dan Hylocereus costaricensis (daging buah super merah). Kulit buah ketiga spesies ini berwarna merah. Bobot per buah berkisar 400g–650g dengan kadar kemanisan mencapai 10–15 Brix. Buah naga termasuk golongan buah non-klimakterik sehingga harus dipanen pada tingkat kematangan yang tepat. Agar diperoleh mutu yang seragam maka setelah buah dipanen dilakukan sortasi. Sortasi secara umum bertujuan menentukan klasifikasi komoditas berdasarkan mutu sejenis yang terdapat dalam komoditas itu sendiri.

2.2. Citra Digital

Citra atau gambar dapat didefinisikan sebagai sebuah fungsi dua dimensi $f(x, y)$ berukuran M baris dan N kolom, dimana x dan y adalah suatu koordinat bidang datar, dan amplitudo f di setiap pasangan koordinat (x, y) disebut intensitas atau level keabuan (grey level) dari suatu citra di titik tersebut. Jika x, y, dan f semuanya mempunyai nilai yang berhingga dan nilainya diskrit, maka citranya disebut citra digital. Sebuah citra digital terdiri dari sejumlah elemen yang berhingga, di mana masing-masing mempunyai lokasi dan nilai tertentu. Elemen-elemen ini disebut sebagai picture element/image element/pels/pixels.

2.2.1. Citra *Greyscale*

Citra *greyscale* merupakan salah satu jenis citra digital yang hanya memiliki satu nilai tertentu pada setiap pikselnya, dengan kata lain nilai RGB-nya sama (*red=green=blue*). Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna hitam, keabuan, dan putih. Tingkatan keabuan (*greyscale level*) yang dimaksud merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih. Banyaknya warna

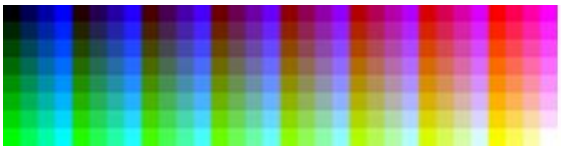
pada citra *greyscale* tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna ini. Citra 2 bit mewakili 4 warna dengan gradasi warna seperti pada Gambar 2.1. Citra 3 bit mewakili 8 warna dengan gradasi warna seperti pada Gambar 1.

Gambar 1. Gradasi warna citra greyscale 2 bit

Gambar 2. Gradasi warna citra greyscale 3 bit

2.2.2. Citra Warna (8 bit)

Setiap piksel pada citra warna (8 bit) mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB atau *Red Green Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna mempunyai gradasi sebanyak 8 bit =28 = 256 warna. Setiap piksel mempunyai kombinasi warna sebanyak 28 .28 .28 = 224 = 16 juta warna lebih. Ada dua jenis citra warna 8 bit. Pertama, citra warna 8 bit dengan menggunakan palet warna 256 dengan setiap paletnya memiliki pemetaan nilai (*colormap*) RGB tertentu. Model ini lebih sering digunakan. Kedua, setiap piksel memiliki format 8 bit yang dinamakan 8 bit *truecolor*.



Gambar 3. Format 8 bit truecolor

2.3. Pengolahan Citra

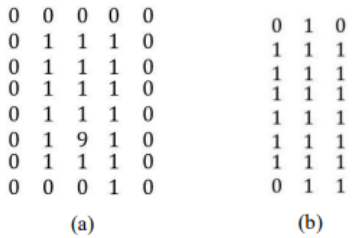
Pengolahan citra adalah pemrosesan gambar dua dimensi dengan menggunakan komputer. Tujuan pengolahan citra digital adalah, yang pertama, untuk memperbaiki citra yang dilihat dari aspek radiometrik (peningkatan kontras, transformasi warna, restorasi citra) dan dari aspek *geometrik* (rotasi, translasi, skala, transformasi *geometrik*). Kedua, untuk melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung dalam citra. Ketiga, untuk melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data. Adapun pengolahan citra yang digunakan dalam penelitian ini sebagai berikut.

2.3.1. Tapis Median (*Median Filter*)

Tapis median merupakan tapis non-linier yang bekerja dengan cara menghitung nilai dari setiap piksel baru, yaitu nilai piksel pada 8 pusat koordinat *sliding window* dengan nilai tengah (median) dari piksel di dalam *window*. Nilai tengah dari piksel di dalam *window* tergantung pada ukuran *sliding window*. Untuk ukuran *window* m baris dan n kolom maka banyaknya piksel dalam window adalah (m x n). Akan lebih baik ukuran window adalah bilangan ganjil karena piksel pada posisi tengahnya lebih pasti diperoleh, yaitu piksel pada posisi (m x n + 1)/2.

Tapis median sangat efektif untuk menghilangkan derau (*noise*) jenis *salt-and-peper* dan juga impuls sementara mempertahankan detail citra karena tidak tergantung pada nilai-nilai yang berbeda dengan nilai-nilai yang umum dalam lingkungannya. Cara kerja tapis median dalam window tertentu adalah dengan mencari nilai mediannya sebagai berikut:

1. Baca nilai piksel yang akan diproses beserta piksel-piksel tetangganya.
2. Urutkan nilai-nilai piksel dari yang paling kecil hingga yang paling besar.
3. Pilih nilai pada bagian tengah untuk nilai yang baru bagi piksel (x, y).



Gambar 4. Citra input, (b) Citra outpoy tapis median 3x3

Gambar 2.4 menunjukkan contoh citra input dan citra output tapis median dengan ukuran window 3x3. Nilai piksel 9 pada citra dianggap derau (memiliki frekuensi tinggi), dengan tapis median derau tersebut hilang pada citra output.

2.3.2. Ekualisasi Histogram Adaptif

Histogram adalah diagram yang menggambarkan frekuensi setiap nilai intensitas yang muncul diseluruh piksel citra. Nilai yang besar menyatakan bahwa nilai intensitas tersebut sering muncul. Ekualisasi histogram memiliki tujuan untuk menghasilkan histogram citra yang seragam. Teknik ini dapat dilakukan pada keseluruhan citra atau pada beberapa bagian saja. Ide dari teknik ini adalah dengan mengubah pemetaan *grey* level agar sebarannya (kontrasnya) lebih luas yaitu pada kisaran 0-255. Sifat dari ekualisasi histogram adalah sebagai berikut.

a. *Grey* level yang sering muncul lebih dijarangkan jaraknya dengan level sebelumnya.
b. *Grey* level yang jarang muncul bisa lebih dirapatkan jaraknya dengan grey level sebelumnya.
c. Histogram baru pasti mencapai nilai maksimal keabuan.

2.4. Data Mining

Data mining adalah proses menganalisis dan mengekstraksi pengetahuan secara otomatis menggunakan satu atau lebih teknik pembelajaran komputer (*machine learning*). Definisi lainnya antara lain pembelajaran berbasis induktif, yaitu proses pembentukan definisi konsep umum yang dilakukan dengan cara mengobservasi contoh-contoh spesifik dari konsep-konsep yang akan dipelajari.

Berikut adalah definisi data mining menurut beberapa ahli beserta sumbernya:

1. Mohammed J. Zaki dan Wagner Meira JR dalam bukunya *"Data Mining and Analysis: Fundamental Concepts and Algorithms*" (2014): *Data mining* adalah proses penemuan pola dalam data. Pola-pola ini harus bermanfaat dan valid, dan tidak seharusnya menjadi acak [2].
2. Menurut Olaode et al. (2014), *data mining* adalah "penggalian informasi atau emas pengetahuan dari sejumlah besar data [3].
3. Menurut Ha et al. (2014), *data mining* adalah "ilmu menemukan informasi baru dengan mencari pola atau aturan dalam data berukuran besar" [4].
4. Menurut Rusdiansyah & Tsaqif (2015), *data mining* adalah "serangkaian proses untuk menggali nilai tambah dalam bentuk pengetahuan yang selama ini tidak diketahui secara manual dari suatu basis data"[5].

2.4.1. Knowledge Discovery in Database

Knowledge Discovery in Database (KDD) adalah proses yang bertujuan untuk menemukan pengetahuan yang berguna dan tersembunyi dari kumpulan data yang besar [6]. KDD sering digunakan dalam konteks *data mining*, di mana teknik dan metode yang berbeda diterapkan untuk mengekstrak pola, tren, dan informasi yang dapat diinterpretasikan dari data.

Proses KDD melibatkan beberapa tahap, yaitu:

1. *Data Selection* (Seleksi): Memilih data yang relevan dari database. Tahap ini melibatkan identifikasi data yang penting dan sesuai untuk dianalisis.
2. *Preprocessing* (Pra-pemrosesan): Melakukan pembersihan data, seperti menghilangkan data yang tidak konsisten, menangani data yang hilang, atau menghapus duplikasi data. Tahap ini bertujuan untuk memastikan kualitas data sebelum analisis dilakukan.
3. *Transformation* (Transformasi): Mengubah data ke dalam bentuk yang sesuai untuk proses penambangan data. Ini bisa mencakup normalisasi data, agregasi, atau transformasi lainnya yang membuat data lebih mudah dianalisis.
4. *Data Mining* (Penambangan Data): Proses inti dari KDD, di mana algoritma dan teknik tertentu diterapkan untuk menemukan pola dan hubungan dalam data. Misalnya, penggunaan algoritma seperti clustering, classification, regression, dan lain-lain.
5. *Interpretation/Evaluation* (Interpretasi/Evaluasi): Menilai dan menafsirkan pola yang ditemukan untuk menentukan apakah pola tersebut valid dan berguna. Hasil dari tahap ini bisa berupa pengetahuan baru yang dapat diterapkan untuk pengambilan keputusan.

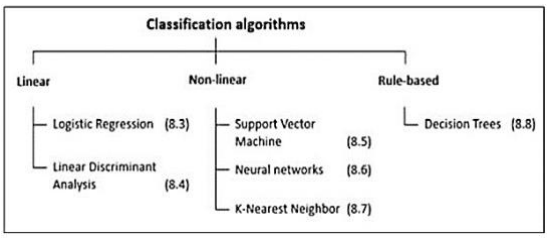
KDD adalah langkah penting dalam analisis data besar karena membantu mengungkapkan informasi yang mungkin tidak terlihat secara eksplisit dalam data mentah.

2.5. Klasifikasi

Berikut penjelasan mengenai klasifikasi menurut ahli beserta sumber kutipannya :

1. Menurut Zhang [7]:

Klasifikasi melibatkan pembelajaran sebuah fungsi yang memetakan data input ke dalam salah satu dari sejumlah kategori diskrit [7]. Terdapat beberapa algoritma dengan pendekatan yang berbeda dalam proses data dan menentukan aturan kriteria pengelompokan klasifikasi. Algoritma ini dibagi menjadi tiga kelompok: berbasis linear, berbasis non-linear, dan rule-based [8].



Gambar 5. Classication algorithms Sumber : Wender, 2016

Jadi klasifikasi secara umum merupakan teknik pemodelan prediktif dengan variable target berupa kategori diskrit, bukan kontinyu. Tujuannya adalah memetakan data input ke dalam kategori diskrit tersebut.

2.6. Penerapan Algoritma K-NN

Algoritma KNN merupakan metode yang digunakan untuk melakukan klasifikasi data berdasarkan jarak terpendek terhadap objek data. Penentuan nilai K yang terbaik untuk algoritma ini berdasarkan pada data yang ada. Nilai K yang tinggi dapat mengurangi efek noise pada klasifikasi, bisa juga membuat batasan antara setiap klasifikasi menjadi lebih kabur [9]. Algoritma KNN merupakan algoritma berbasis contoh atau non parametric dan dianggap metode paling sederhana di dalam proses *data mining.*

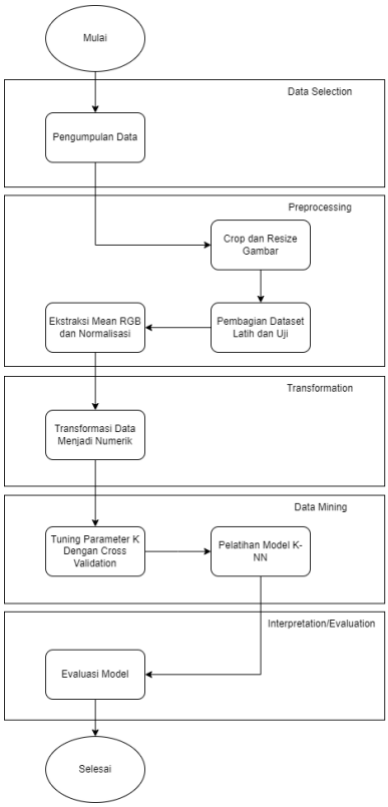
Algoritma KNN salah satu metode klasifikasi data yang mudah diimplementasi pada jumlah data yang kecil, tetapi jika *dataset* yang diolah banyak dan kompleks maka algoritma KNN memiliki kelemahan dan waktu yang tidak efisien. Metode KNN menggunakan ukuran jarak yang sesuai untuk mengklasifikasikan data baru. Jarak tetangga terdekat K dihitung dan label kelas dari tetangga terdekat diprediksi sebagai label kelas dari instance baru. Akurasi KNN sangat terpengaruh dengan memilih jumlah K tetangga terdekat. Jika terdapat nilai K yang kecil, maka akan sensitif terhadap noise dan jika terlalu besar, dapat menyebabkan bias model.

Algoritma KNN merupakan algoritma berbasis memori yang menggunakan iterasi pada data sampai atribut atau parameter data yang terdekat ditemukan. Jarak minimum data yang diproses pada data testing akan dibandingkan dengan data training dengan jarak yang terdekat. Algoritma KNN digunakan untuk memprediksi harga jual tanah dengan cara menentukan pilihan dalam proses pemilihan tanah yang strategis dengan harga yang sesuai

Algoritma KNN digunakan untuk prediksi penjualan furniture sehingga dapat mengurangi waktu pemrosesan dan algoritma tersebut bisa memberikan akurasi yang baik dalam sampel pengujian yang telah. Algoritma KNN digunakan untuk mengklasifikasikan pemasaran dengan hasil dan akurasi menggunakan metode *cross validation* dengan hasil yang baik.

3. METODE PENELITIAN

Berikut ini adalah tahapan dari penelitian pada Gambar 6



Gambar 6. Tahapan Penelitian

3.1. Pengumpulan Data

Pengumpulan data dilakukan melalui observasi langsung di lapangan, di mana buah naga dinilai berdasarkan kriteria seperti warna kulit, ukuran, dan tekstur. Setiap data dicatat dengan teliti untuk memastikan keakuratan dan konsistensi dalam proses klasifikasi yang akan datang. Kriteria ini dipilih berdasarkan relevansi terhadap kematangan buah naga, yang merupakan faktor kunci dalam menentukan apakah buah tersebut sudah layak panen atau masih memerlukan waktu untuk masak.

3.2. Preprocessing

3.2.1. Crop dan Resize Gambar

Proses *crop* dan *resize* gambar ini bertujuan untuk mengubah semua gambar menjadi berukuran 100x100 piksel.

3.2.2. Pembagian Dataset

Dalam proses pengembangan model *k-nearest neighbor*, *dataset* yang telah terkumpul data dibagi menjadi dua bagian utama: data latih dan data uji. Pembagian ini dilakukan dengan rasio 60% untuk data latih dan 40% untuk data uji.

3.2.3. Ekstrasi Mean RGB dan Normalisasi

Proses ekstraksi *mean* RGB dan normalisasi gambar dalam konteks pengolahan citra melibatkan beberapa langkah penting untuk mempersiapkan data sebelum analisis lebih lanjut atau pelatihan model. Salah satu langkah yang sering dilakukan adalah menghitung rata-rata nilai RGB dari setiap piksel gambar, diikuti dengan normalisasi *min-max* pada data tersebut.

Setelah memperoleh nilai rata-rata RGB, langkah berikutnya adalah melakukan normalisasi *min-max*. Normalisasi ini bertujuan untuk mengubah rentang nilai data ke dalam rentang yang konsisten, biasanya antara 0 dan 1. Proses ini dilakukan dengan mengurangi nilai minimum dari data dan membaginya dengan rentang nilai (selisih antara nilai maksimum dan minimum). Dengan melakukan normalisasi min-max pada nilai rata-rata RGB, data akan menjadi lebih seragam dan lebih mudah digunakan dalam model machine learning atau analisis statistik. Normalisasi ini juga membantu dalam mengurangi dampak perbedaan skala dan pencahayaan antara gambar yang berbeda, sehingga model dapat belajar dengan lebih efektif dan akurat.

3.3. Transformation

3.3.1. Transformasi Label Data Menjadi Numerik

Transformasi label data menjadi numerik adalah proses yang sangat penting dalam pemrosesan data. Banyak algoritma KNN yang memerlukan input data dalam bentuk numerik agar dapat melakukan perhitungan matematika yang diperlukan. Karena itu, label data yang berbentuk kategori atau teks harus diubah menjadi angka sebelum dapat digunakan dalam model.

Salah satu metode yang umum digunakan untuk transformasi ini adalah label *encoding*. Dalam label *encoding*, setiap kategori dalam data diberikan angka unik. Transformasi label data menjadi numerik ini sangat penting karena memungkinkan model machine learning untuk memproses dan memahami data dengan lebih baik, serta meningkatkan akurasi dan efektivitas model dalam membuat prediksi atau keputusan.

3.4. Data Mining

3.4.1. Tuning Paramenter *K* Dengan *Cross Validation*

Tuning parameter *k* dalam algoritma *K-Nearest Neighbors* (KNN) menggunakan cross-validation adalah proses krusial untuk mengoptimalkan kinerja model. Proses ini dimulai dengan membagi data menjadi beberapa subset menggunakan teknik cross-validation, seperti *k-fold cross-validation*. Setiap *subset*, atau *fold*, digunakan bergantian sebagai data validasi sementara sisanya menjadi data pelatihan.

Dengan menggunakan nilai *k* yang telah dioptimalkan, model KNN dilatih ulang menggunakan seluruh data pelatihan untuk memaksimalkan kemampuannya dalam membuat prediksi yang akurat pada data baru. Proses ini membantu memastikan bahwa model KNN memiliki keseimbangan yang tepat antara bias dan variansi, serta dapat melakukan generalisasi dengan lebih baik pada data yang belum terlihat.

3.4.2. Pelatihan Model K-NN

Pelatihan model *K-Nearest Neighbors* (KNN) dengan nilai *k* terbaik adalah proses akhir dari *tuning* parameter yang bertujuan untuk mengoptimalkan kinerja model. Setelah menentukan nilai *k* yang memberikan hasil terbaik melalui *cross-validation*, langkah selanjutnya adalah melatih model KNN dengan nilai *k* tersebut menggunakan seluruh data pelatihan.

Ketika model KNN yang telah dilatih harus membuat prediksi untuk data baru, ia menghitung jarak antara data baru dan semua titik dalam dataset pelatihan. Jarak ini bisa dihitung menggunakan matrik jarak *Euclidean*. Model kemudian memilih *k* tetangga terdekat berdasarkan jarak yang dihitung dan menentukan kelas atau nilai prediksi dengan mayoritas suara dari tetangga-tetangga tersebut.

Pelatihan dengan nilai *k* terbaik memastikan bahwa model memanfaatkan parameter yang optimal untuk keseimbangan antara sensitivitas terhadap *noise* dan kemampuan untuk menggeneralisasi. Dengan menggunakan nilai *k* yang sudah teruji memberikan hasil terbaik pada data validasi, model KNN diharapkan dapat membuat prediksi yang lebih akurat dan lebih dapat diandalkan saat dihadapkan dengan data yang belum terlihat sebelumnya.

3.5. Interpretation/Evaluation

a. Evaluasi Model

Untuk setiap nilai K, model K-NN dilatih menggunakan data latih dan dievaluasi menggunakan data uji. Metrik evaluasi seperti akurasi, *precision*, *recall*, dan *F1-score* dihitung untuk masing-masing nilai K.

b. Akurasi

Akurasi adalah proporsi prediksi yang benar dari total jumlah prediksi. Akurasi dihitung dengan membagi jumlah prediksi benar (baik positif maupun negatif) dengan jumlah total data. Akurasi memberikan indikasi umum tentang kinerja model, namun metrik ini bisa menyesatkan jika *dataset* tidak seimbang. Misalnya, jika ada 90% data dari satu kelas, model yang selalu memprediksi kelas mayoritas akan memiliki akurasi tinggi, meskipun kinerjanya tidak baik dalam mengenali kelas minoritas.

c. Precision

Precision adalah proporsi prediksi positif yang benar dari semua prediksi positif. *Precision* memberikan gambaran tentang seberapa akurat prediksi positif model. *Precision* tinggi berarti dari semua prediksi positif yang dibuat model, sebagian besar adalah benar. *Precision* sangat penting ketika biaya untuk kesalahan positif (*false positive*) tinggi, seperti dalam kasus deteksi penipuan.

d. Recall

Recall, juga dikenal sebagai sensitivitas atau *true positive rate*, adalah proporsi kasus positif yang benar-benar terdeteksi oleh model dari semua kasus positif yang sebenarnya. *Recall* menunjukkan seberapa baik model dalam menemukan semua contoh positif dalam *dataset*. *Recall* tinggi berarti model berhasil menangkap sebagian besar dari data positif. *Recall* sangat penting ketika kesalahan negatif (*false negative*) tidak dapat diterima, seperti dalam deteksi penyakit.

e. F1-Score

F1-Score adalah rata-rata harmonis dari *precision* dan *recall*, yang memberikan keseimbangan antara keduanya. *F1-Score* digunakan sebagai metrik tunggal ketika penting untuk mempertimbangkan *precision* dan *recall* secara bersamaan, terutama ketika ada *trade-off* antara keduanya. *F1-Score* memberikan indikasi yang lebih baik tentang kinerja model ketika dataset tidak seimbang atau ketika penting untuk meminimalkan kedua jenis kesalahan (*false positives* dan *false negatives*).

4. HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data

Proses pengumpulan data untuk pengembangan model K-NN dalam klasifikasi buah naga dilakukan dengan tujuan mengelompokkan buah naga ke dalam tiga kategori: "layak panen," "matang," dan "mentah." Sebanyak 256 titik data telah berhasil dikumpulkan, mencakup berbagai parameter yang relevan untuk penentuan kategori tersebut.

Tabel 1. Pembagian Data Latih Uji

Data Layak Panen	Data Matang	Data Mentah
70 Buah	124 Buah	70 Buah

4.2. *Pre-proccesing* Data

Pada tahap ini, gambar yang ada di folder input akan diproses untuk mendapatkan ukuran yang seragam. Proses pertama adalah melakukan *crop* untuk memastikan bahwa gambar yang diproses berbentuk persegi. Ini dicapai dengan menentukan area *crop* yang merupakan bagian tengah dari gambar asli. Ukuran *crop* ditentukan berdasarkan dimensi terkecil dari gambar asli, memastikan bahwa *crop* dilakukan secara simetris dari tengah. Setelah gambar di-*crop* menjadi bentuk persegi, gambar tersebut kemudian di-*resize* ke ukuran yang telah ditentukan, yaitu 100x100 piksel. Proses ini memastikan bahwa semua gambar memiliki dimensi yang konsisten, yang penting untuk model klasifikasi agar tidak terpengaruh oleh variasi ukuran gambar.

Hasil akhir dari proses ini adalah gambar yang telah di-*crop* dan di-*resize* disimpan di folder tujuan, siap digunakan untuk tahap berikutnya dalam pelatihan model.

Berikut adalah contoh hasilnya:

Gambar 7. Gambar sebelum Crop dan Resize

Gambar 8. Gambar Setelah Crop dan Resize

Setelah dilakukan *croping dan resize* Hasil dari pembagian dataset ini adalah sebagai berikut: untuk kategori "layak panen," terdapat 42 data yang digunakan sebagai data latih dan 28 data sebagai data uji. Pada kategori "matang" 74 data digunakan sebagai data latih, sedangkan 50 data digunakan sebagai data uji. Terakhir, untuk kategori "mentah" 42 data dialokasikan untuk data latih dan 28 data untuk data uji.

Data Latih Layak Panen	Data Latih Matang	Data Latih Mentah	Data Uji Layak Panen	Data Uji Matang	Data Uji Mentah
42 Buah	74 Buah	42 Buah	28 Buah	50 Buah	28 Buah

Setelah data gambar buah naga, selanjutnya dilakukan Ekstrasi *Mean* RGB dan Normalisasi

mean_r	mean_g	mean_b	kelas
0,306616	0,496252	0,09338	0
0,464766	0,645501	0,159657	0
0,257645	0,52063	0,028373	0
0,292046	0,496653	0,151474	0
0,285882	0,491788	0,119992	0
0,28401	0,489954	0,12051	0
0,282895	0,542064	0,050787	0
0,307035	0,498163	0,091502	0
0,279767	0,488367	0,118132	0
0,236916	0,482349	0,047116	0
0,289009	0,494814	0,128679	0
0,263704	0,509436	0,104807	0
0,345657	0,601778	0,072261	0
0,243089	0,48718	0,053552	0
0,298745	0,502264	0,138951	0
0,292937	0,495478	0,146717	0
0,338139	0,582804	0,073497	0
0,232791	0,480288	0,037815	0
0,34335	0,522079	0,125085	0
0,303718	0,493191	0,090148	0
0,297334	0,499542	0,150627	0
0,311639	0,509212	0,170759	0
0,336159	0,522567	0,054585	0
0,301221	0,491661	0,087193	0
0,353318	0,529934	0,13823	0
0,275884	0,476305	0	0
0,252035	0,436106	0,188732	0
0,317329	0,55657	0,083596	0
0,301042	0,487851	0,143306	0
0,215523	0,475026	0,03136	0
0,289195	0,478536	0,135366	0
0,350569	0,592648	0,079042	0
0,301837	0,484315	0,143667	0
0,278592	0,485025	0,131059	0

Gambar 9. Nilai Ekstrasi Mean setelah dinomalisasi

Pada fungsi normalisasi *Min-Max* diperkenalkan untuk menskalakan data sehingga berada dalam rentang nilai tertentu. Ini merupakan langkah penting dalam pra-pemrosesan data sebelum pelatihan model untuk memastikan bahwa semua fitur berada pada skala yang sama.

4.3. Transformasi Label Data Menjadi Numerik

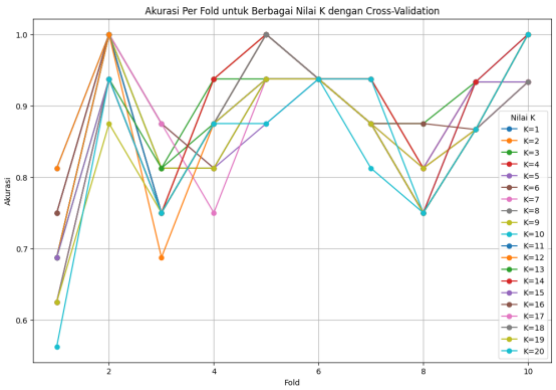
Evaluasi dilakukan dengan menggunakan data. Langkah ini memastikan bahwa setiap kelas memiliki label numerik yang unik, dalam kasus ini Layak Panen menjadi 0, Matang menjadi 1, dan Mentah menjadi 2, yang kemudian digunakan sebagai target dalam pelatihan model *machine learning*.

Pemetaan label kelas ke bentuk numerik sangat penting dalam konteks *machine learning* karena sebagian besar algoritma hanya dapat memproses data numerik. Oleh karena itu, proses ini menjadi krusial dalam memastikan data siap digunakan untuk pelatihan model, sehingga model dapat mengenali dan membedakan kelas-kelas yang ada berdasarkan angka, bukan teks. Dengan demikian, proses klasifikasi dapat dilakukan secara efektif dan akurat.

4.4. Data Mining

4.4.1. Tunning Parameter K dengan Cross Validation

Pada kode ini, proses pemilihan nilai terbaik untuk parameter K pada algoritma *K-Nearest Neighbors* (KNN) dilakukan menggunakan teknik *cross-validation* Pertama, rentang nilai K yang akan diuji ditentukan, yaitu dari 1 hingga 20. Untuk setiap nilai K, proses *cross-validation* dilakukan untuk mengukur kinerja model. Jumlah *fold* untuk *cross-validation* diatur menjadi 10, dengan *KFold* dari *scikit-learn* yang digunakan untuk membagi data latih menjadi 10 bagian, di mana setiap bagian secara bergantian digunakan sebagai data uji, sementara sisanya digunakan untuk pelatihan.



Gambar 10. Tunning Ten-Fold Validation

Berikut adalah hasil akurasi tiap nilai K :

```
Akurasi rata-rata untuk K = 1: 89.88%
Akurasi rata-rata untuk K = 2: 89.88%
Akurasi rata-rata untuk K = 3: 89.92%
Akurasi rata-rata untuk K = 4: 88.67%
Akurasi rata-rata untuk K = 5: 88.04%
Akurasi rata-rata untuk K = 6: 88.63%
Akurasi rata-rata untuk K = 7: 86.75%
Akurasi rata-rata untuk K = 8: 86.12%
Akurasi rata-rata untuk K = 9: 87.42%
Akurasi rata-rata untuk K = 10: 87.42%
Akurasi rata-rata untuk K = 11: 87.46%
Akurasi rata-rata untuk K = 12: 86.79%
Akurasi rata-rata untuk K = 13: 87.42%
Akurasi rata-rata untuk K = 14: 87.46%
Akurasi rata-rata untuk K = 15: 86.79%
Akurasi rata-rata untuk K = 16: 86.17%
Akurasi rata-rata untuk K = 17: 85.54%
Akurasi rata-rata untuk K = 18: 86.17%
Akurasi rata-rata untuk K = 19: 84.92%
Akurasi rata-rata untuk K = 20: 83.67%
Nilai K terbaik: 3 dengan akurasi rata-rata: 89.92%
```

Gambar 11. Hasil Output Proses Cross Validation

4.4.2. Pelatihan Model K-NN

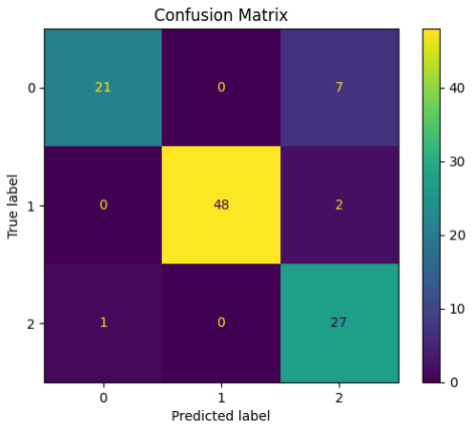
Pada tahap ini, model *K-Nearest Neighbors* (KNN) dilatih menggunakan nilai K terbaik yang telah ditentukan dari proses *tuning* sebelumnya. Proses dimulai dengan menginisialisasi objek model KNN dengan parameter *n_neighbors* yang diset ke

nilai *best_k*, yakni nilai K yang memberikan performa terbaik berdasarkan evaluasi *cross-validation*. Setelah inisialisasi, model dilatih dengan data pelatihan yang telah dinormalisasi, yaitu *X_train_scaled* sebagai fitur dan *y_train* sebagai label.

4.5. Interpretation/Evaluation

Hasil prediksi dibandingkan dengan label asli untuk menghitung metrik evaluasi utama, termasuk akurasi, *precision*, *recall*, dan *F1 score*. Metrik ini memberikan gambaran menyeluruh tentang kinerja model dalam klasifikasi data. Selain itu, *confusion matrix* dihitung untuk menilai performa model secara lebih mendetail, dengan menampilkan jumlah prediksi yang benar dan salah untuk setiap kelas.

Hasil evaluasi, termasuk *confusion matrix* dan *classification report*, ditampilkan untuk memberikan wawasan mendalam tentang kemampuan model dalam klasifikasi. Sebagai langkah akhir, data asli dan hasil prediksi diekspor ke file Excel, yang memudahkan dokumentasi dan analisis lebih lanjut. Untuk membantu dalam visualisasi hasil, confusion matrix juga ditampilkan menggunakan *ConfusionMatrixDisplay*, memberikan representasi grafis dari hasil klasifikasi.



Gambar 12. Output Program Perhitungan Confusion Matrix

Pada matriks ini, setiap kotak mewakili jumlah prediksi yang sesuai antara kelas sebenarnya dengan kelas yang diprediksi. Warna pada kotak menunjukkan frekuensi prediksi, di mana warna yang lebih terang menandakan jumlah prediksi yang lebih tinggi.

Misalnya, kelas '1' terlihat memiliki performa yang sangat baik dengan 48 prediksi benar dan hanya 2 prediksi salah (diprediksi sebagai kelas '2'). Kelas '0' dan '2' juga menunjukkan performa yang baik dengan sebagian besar prediksi benar, meskipun terdapat beberapa kesalahan prediksi.

Visualisasi *confusion matrix* ini membantu dalam memahami kesalahan model dengan lebih baik dan menunjukkan di mana model mungkin perlu ditingkatkan, seperti dengan mengurangi kesalahan prediksi di antara kelas-kelas tertentu.

4.6. Knowledge

Penelitian ini memberikan pengetahuan yang berharga di beberapa bidang. Pertama, penelitian ini menunjukkan efektivitas algoritma *K-Nearest Neighbors* (K-NN) dalam mengklasifikasikan tingkat kematangan buah naga berdasarkan warna kulitnya dengan akurasi yang tinggi, tanpa memerlukan pemeriksaan fisik atau pemotongan buah.

Selain itu, studi ini menyoroti penerapan proses *Knowledge Discovery in Databases* (KDD) dalam analisis data pertanian, terutama dalam memproses dan menganalisis data citra buah. Penelitian ini menekankan pentingnya setiap langkah dalam KDD, mulai dari pemilihan data, praproses, transformasi, hingga penerapan algoritma pembelajaran mesin untuk menghasilkan wawasan yang bermanfaat.

Penelitian ini juga berkontribusi pada bidang pertanian presisi dengan menawarkan metode untuk menentukan kematangan buah naga secara otomatis, yang dapat mengoptimalkan waktu panen dan meningkatkan kualitas hasil panen secara keseluruhan.

Terakhir, penelitian ini menegaskan pentingnya model warna RGB dalam ekstraksi fitur, serta bagaimana data warna dapat dimanfaatkan untuk membuat keputusan yang lebih baik dalam praktik pertanian

5. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem klasifikasi tingkat kematangan buah naga berdasarkan warna kulit menggunakan algoritma *K-Nearest Neighbors* (K-NN). Dari hasil analisis dan implementasi, dapat disimpulkan bahwa algoritma K-NN mampu mengklasifikasikan tingkat kematangan buah naga dengan akurasi yang tinggi, terutama pada nilai K tertentu. Akurasi terbaik diperoleh pada nilai K = 3 dengan persentase akurasi sebesar 91%. Fitur warna RGB *(Red, Green, Blue*) terbukti efektif dalam membedakan tingkat kematangan buah naga. Nilai rata-rata (*mean*) dari ketiga komponen warna ini menjadi input yang penting dalam model klasifikasi. Penggunaan *Min-Max Normalization* membantu dalam menstandarisasi data sehingga semua fitur berada pada skala yang sama, yang sangat penting untuk kinerja model K-NN. *Scatter plot* dari data latih dan data uji menunjukkan distribusi yang jelas antara kelas matang dan mentah, sementara *Confusion Matrix* membantu dalam mengevaluasi performa model dengan memperlihatkan jumlah prediksi benar dan salah untuk setiap kelas. Aplikasi berbasis GUI menggunakan *Tkinter* dan PIL memudahkan pengguna untuk melakukan prediksi secara langsung dengan mengunggah gambar buah naga. Aplikasi ini juga menyimpan hasil prediksi dan jarak Euclidean ke file Excel, memungkinkan analisis lebih lanjut.

6. DAFTAR PUSTAKA

[1] Budianto, A., Ariyuana, R., & Maryono, D. (2018). *Perbandingan K-Nearest Neighbor* (KNN) dan *Support Vector Machine (SVM)* untuk Klasifikasi Data.

[2] Mohammed, J. Z., & Wagner, M. JR. (2014). *Data Mining and Analysis: Fundamental Concepts and Algorithms*

[3] Olaode, A. A., Adeyemo, N. G., & Gbadamosi, T. C. A. (2014). Unstructured data mining for smart remote monitoring and diagnostic of industrial plant. In 2014 IEEE Conference on Control Applications (CCA) (pp. 1920-1925). IEEE.

[4] Ha, K., Ahn, C., & Lee, B. J. (2014). Data-driven soft sensor approach for quality prediction in steel industry. IEEE Transactions on Industrial Informatics, 10(1), 111–118.

[5] Rusdiansyah, A., & Taufik, M. (2015). Implementasi data mining untuk evaluasi kinerja akademik mahasiswa menggunakan algoritma *Naive Bayes classifier*. Jurnal Ilmiah FIFO

[6] Mubarok, M. I., Purwantoro, & Carudin. (2023). Penerapan algoritma *k-nearest neighbor* (KNN) dalam klasifikasi penilaian jawaban ujian esai. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(5), 3446.

[7] Zhang, Y., Yun, Y., An, R., Cui, J., Dai, H., & Shang, X. (2021). Educational Data Mining Techniques for Student Performance Prediction: Method Review and Comparison Analysis. Frontiers in Psychology, 12. https://doi.org/10.3389/fpsyg.2021.698490.

[8] Pradytya, D. A. (2018). Kajian *Data Mining* Untuk Memprediksi Kelulusan Mahasiswa Dengan Metode Klasifikasi (Studi Kasus: STMIK LIKMI Bandung).

[9] Fadillah, M. F. A., & Mutawakkil, M. R. N. (2023). K-Nearest Neighbor Untuk Klasifikasi Jenis Buah Berdasarkan Berat, Tinggi, dan Lebar.