

**MODIFIKASI METODE K-NEAREST NEIGHBOR DENGAN
NORMALIZED EUCLIDEAN DISTANCE UNTUK KLASIFIKASI
KUALITAS JERUK LOKAL BERASTAGI**

DISUSUN OLEH

ANANDA AFIFAH SIREGAR

2009020098



PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER & TEKNOLOGI INFORMASI

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

MEDAN

2024

**MODIFIKASI METODE K-NEAREST NEIGHBOR DENGAN
NORMALIZED EUCLIDEAN DISTANCE UNTUK KLASIFIKASI
KUALITAS JERUK LOKAL BERASTAGI**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer
(S.Kom) dalam Program Studi Teknologi Informasi pada Fakultas Ilmu Komputer
dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara**

ANANDA AFIFAH SIREGAR

NPM. 2009020098

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER & TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2024

LEMBAR PENGESAHAN

Judul Skripsi : Modifikasi Metode K-Nearest Neighbor Dengan Normalized Euclidean Distance Untuk Klasifikasi Kualitas Jeruk Lokal Berastagi
Nama Mahasiswa : Ananda Afifah Siregar
NPM : 2009020098
Program Studi : Teknologi Informasi

**Menyetujui
Komisi Pembimbing**



(Dr. Al-Khowarizmi, S.Kom., M.Kom)
NIDN. 0127099201

Ketua Program Studi



(Fatma Sari Hutagalung, S.Kom., M.Kom)
NIDN. 0117019301

Dekan



(Dr. Al-Khowarizmi, S.Kom., M.Kom)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

**MODIFIKASI METODE K-NEAREST NEIGHBOR DENGAN
NORMALIZED EUCLIDEAN DISTANCE UNTUK KLASIFIKASI
KUALITAS JERUK LOKAL BERASTAGI**

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 1 November 2024

Yang membuat pernyataan



Ananda Afifah Siregar

NPM. 2009020098

**PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Ananda Afifah Siregar
NPM : 2009020098
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bebas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

**Modifikasi Metode K-Nearest Neighbor Dengan Normalized Euclidean Distance
Untuk Klasifikasi Kualitas Jeruk Lokal Berastagi**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 1 November 2024

Yang membuat pernyataan



Ananda Afifah Siregar

NPM. 2009020098

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Ananda Afifah Siregar
Tempat dan Tanggal Lahir : Ingul Jae, 02 November 2002
Alamat Rumah : Desa Ingul Jae,
Telepon/Faks/HP : 082165445768
E-mail : afifahananda5@gmail.com
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD : SDN 102630 Ingul Jae TAMAT: 2014
SMP : MTsN 1 Model Padang Sidempuan TAMAT: 2017
SMA : MAN 2 Model Medan TAMAT: 2020

KATA PENGANTAR



Alhamdulillah, segala puji syukur bagi Allah SWT yang telah memberikan rahmat dan karunia-Nya sehingga penulis mendapatkan kemudahan dalam menyelesaikan skripsi ini yang berjudul *“Modifikasi Metode K-Nearest Neighbor Dengan Normalized Euclidean Distance Untuk Klasifikasi Kualitas Jeruk Lokal Berastagi”* dimana skripsi ini sangat penulis butuhkan dalam rangka sebagai kelengkapan penulis untuk memperoleh gelar sarjana Komputer pada Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara. Penulis berterima kasih kepada berbagai pihak yang tidak terlepas dari dukungan serta doa dalam penyelesaian skripsi ini. Oleh karena itu, penulis ingin menyampaikan terima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., selaku Rektor Universitas Muhammadiyah Sumatera Utara.
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom, selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara serta dosen pembimbing yang telah memberikan arahan dan bimbingan kepada penulis.
3. Bapak Halim Maulana, ST, M.Kom, selaku Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara.
4. Bapak Lutfi Basit, S.Sos, M.I.Kom, selaku Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi Muhammadiyah Sumatera Utara.

5. Ibu Fatma Sari Hutagalung, S.Kom, M.Kom, selaku Ketua Program Studi Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara.
6. Bapak Mhd. Basri, S.Si, M.Kom selaku Sekretaris Program Studi Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara.
7. Bapak dan Ibu dosen Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara yang telah memberikan banyak bekal ilmu kepada penulis.
8. Seluruh pegawai Biro Administrasi Teknologi Informasi Universitas Muhammadiyah Sumatera Utara yang telah membantu dalam menyelesaikan segala berkas administrasi yang dibutuhkan selama perkuliahan.
9. Kedua orang tua penulis, Lukman Siregar dan Jumrotul Haida Nst serta saudara-saudari penulis yang telah memberikan doa dan dukungan kepada penulis sehingga dapat menyelesaikan skripsi ini.
10. Dan teman-teman seperjuangan yang telah membantu saya dalam penyusunan skripsi ini terutama seluruh teman-teman Teknologi Informasi stambuk 2020 yang telah banyak memberikan bantuan serta dukungannya kepada penulis

ABSTRAK

Buah lokal Indonesia merupakan salah satu contoh kekayaan alam yang dimiliki negara ini, salah satu nya adalah jeruk lokal Berastagi. Jeruk kaya akan vitamin C yang bermanfaat bagi kesehatan tubuh. Rasanya cenderung asam, segar, dan manis. Kandungan vitamin C dalam jeruk mencapai 97,3 miligram atau setara dengan 163 % dari angka kecukupan gizi. Selain vitamin C, buah jeruk juga mengandung vitamin B6, antioksidan dan serat. Oleh sebab itu, sangat disarankan untuk mengonsumsi buah jeruk setiap hari karena dapat membantu melancarkan pencernaan, mengurangi resiko diabetes, menjaga kesehatan kulit, serta menjaga daya tahan tubuh. Penelitian ini bertujuan untuk menerapkan Klasifikasi dan penilaian kualitas jeruk lokal dengan menggunakan metode K-Nearest Neighbor (KNN) yang dimodifikasi dengan Normalized Euclidean distance untuk meng-klasifikasikan kualitas jeruk lokal Berastagi berdasarkan warna citra buah. Dataset penelitian ini diambil dari gambar buah jeruk lokal Berastagi sebanyak 100 gambar, dimana 100 gambar tersebut dibagi menjadi 2 yaitu, jeruk baik dan jeruk buruk. Proses klasifikasi buah jeruk lokal Berastagi menggunakan aplikasi matlab.

Kata Kunci: Klasifikasi, Jeruk Lokal Berastagi, K-Nearest Neighbor, Normalized Euclidean Distance, Matlab.

ABSTRACT

Local Indonesian fruits are one example of the natural wealth owned by this country, one of which is the local Berastagi orange. Oranges are rich in vitamin C which is beneficial for body health. The taste tends to be sour, fresh, and sweet. The vitamin C content in oranges reaches 97.3 milligrams or equivalent to 163% of the nutritional adequacy rate. In addition to vitamin C, oranges also contain vitamin B6, antioxidants and fiber. Therefore, it is highly recommended to consume oranges every day because it can help smooth digestion, reduce the risk of diabetes, maintain healthy skin, and maintain endurance. This study aims to apply the Classification and assessment of the quality of local oranges using the K-Nearest Neighbor (KNN) method modified with Normalized Euclidean distance to classify the quality of local Berastagi oranges based on the color of the fruit image. The research dataset was taken from 100 images of local Berastagi oranges, where the 100 images were divided into 2, namely, good oranges and bad oranges. The classification process of local Berastagi oranges uses the matlab application.

Keywords: Classification, Berastagi Local Orange, K-Nearest Neighbor, Normalized Euclidean Distance, Matlab.

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN ORISINALITAS.....	ii
PERNYATAAN PERSETUJUAN PUBLIKASI.....	iii
RIWAYAT HIDUP.....	iv
KATA PENGANTAR.....	v
ABSTRAK	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
BAB I. PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	4
1.3. Batasan Masalah.....	4
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian	5
BAB II. LANDASAN TEORI	6
2.1. Data Mining	6
2.2. Klasifikasi.....	6
2.3. <i>Algoritma K-Nearest Neighbor (KNN)</i>	7
2.4. Jeruk Lokal	8
2.5. Matlab.....	9
2.6. Flowchart	10
2.7. UML.....	11
BAB III. METODOLOGI PENELITIAN	15
3.1. Metode Penelitian	15
3.2. Dataset	15
3.3. Normalized	17
3.4. Prosedur Penelitian	18

3.5. Alur Perancangan Sistem	20
3.6. Flowchart Sistem Klasifikasi	22
3.7. Unified Modeling Language (UML) Diagram	23
3.8. Use Case Diagram	26
3.9. Activity Diagram.....	27
3.10. Sequence Diagram.....	28
3.11. Antar Muka Sistem.....	29
BAB IV. HASIL DAN PEMBAHASAN	30
4.1. Tampilan Sistem	30
4.2. Nilai RGB (<i>Red, Green, Blue</i>)	31
4.3. <i>Euclidean Distance</i>	33
4.4. <i>Normalized Euclidean Distance</i>	35
4.5. Hasil Pengujian.....	38
BAB V. KESIMPULAN DAN SARAN.....	46
5.1. Kesimpulan	46
5.2. Saran.....	47
DAFTAR PUSTAKA.....	36

DAFTAR TABEL

		HALAMAN
Tabel 2.1.	Simbol Flowchart	10
Tabel 3.1.	Dataset	15
Tabel 4.1	Nilai RGB Sampel Jeruk	32
Tabel 4.2	Sampel Yang Akan Diuji	33
Tabel 4.3	Euclidean Distance	34
Tabel 4.4	Nilai Yang Akan Diubah	35
Tabel 4.5	Nilai Normalized RGB	36
Tabel 4.6	Normalized Euclidean Distance	38
Tabel 4.7	Hasil Pengujian	39

DAFTAR GAMBAR

	HALAMAN
Gambar 2.1. <i>Algoritma K-Nearest Neighbor (KNN)</i>	8
Gambar 2.2 Jeruk Lokal Berastagi	9
Gambar 2.3 Simbol <i>Use Case Diagram</i>	12
Gambar 2.4 Simbol Activity Diagram	13
Gambar 2.5 Sequence Diagram	14
Gambar 3.1 Contoh Normalisasi Buah Jeruk	18
Gambar 3.2 Proses Tahapan Penelitian	19
Gambar 3.3 Alur Perancangan Sistem	21
Gambar 3.4 Flowchart Sistem Klasifikasi	22
Gambar 3.5 UML Diagram	23
Gambar 3.6 Use Case Diagram	26
Gambar 3.7 Activity Diagram	27
Gambar 3.8 Sequence Diagram	28
Gambar 3.9 Tampilan Antar Muka Sistem	29
Gambar 4.1 Tampilan Sistem	30
Gambar 4.2 Proses Nilai RGB	31
Gambar 4.3 Contoh Pengujian Sampel 3	32

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Sejalan dengan kemajuan ilmu dan teknologi informasi saat ini, yang mengalami pertumbuhan yang sangat cepat (Fauzi, E., et al. 2022). Semakin terdorong untuk memanfaatkan teknologi canggih sebagai sarana untuk menjaga keberadaannya di masyarakat (Misar & Prasetya, A.R., 2015). Dampak dari semakin pesatnya teknologi informasi manusia dapat menentukan kualitas jeruk lokal dengan adanya perancangan sistem klasifikasi kualitas buah jeruk dengan memanfaatkan teknologi modern yang telah berkembang dengan cepat.

Data mining ialah sebuah tahapan yang bertujuan untuk diketahui oleh pengguna dan menyajikannya menggunakan cara yang sederhana agar mudah dimengerti oleh pengguna sehingga hubungan itu menjadi landasan dalam pengambilan Keputusan (Ridwan, M., et al. 2013). Selama ini, proses pengklasifikasian kualitas buah jeruk masih dilakukan secara manual dengan mengamati bagian luar buah, sehingga membuat klasifikasi secara manual kurang efektif karena terdapat kesalahan yang disebabkan oleh manusia. Oleh sebab itu, dibutuhkannya sistem klasifikasi kualitas buah jeruk agar hasil pengklasifikasian lebih akurat. Proses pengklasifikasian yang akurat dapat dilihat dari beberapa ciri seperti warna buah.

Terdapat berbagai metode atau algoritma dalam *data mining* yang dapat melakukan proses klasifikasi, diantaranya ialah algoritma *K-Nearest Neighbor* (KNN) (Fajarini,

I.P., et al. 2022). Algoritma *K-Nearest Neighbor* (KNN) ialah salah satu metode yang dapat diterapkan untuk mengklasifikasikan suatu objek berdasarkan data yang paling mirip dengan objek tersebut (Kasanah, A.N., et al. 2019). *K-Nearest Neighbors* (KNN) adalah metode pembelajaran mesin yang memanfaatkan data sebelumnya untuk memprediksi hasil data yang akan datang. Algoritma *K-Nearest Neighbor* selain digunakan untuk peramalan, juga berfungsi untuk memprediksi klasifikasi. KNN adalah pengembangan lebih lanjut dari teknik klasifikasi *Nearest Neighbor*. Ini terbukti karena setiap data baru dapat dianalisis berdasarkan k tetangga terdekatnya, di mana k merupakan bilangan bulat positif, bahkan dengan dataset yang kecil (Al-Khowarizmi., et al. 2021).

K-Nearest Neighbor (KNN) dalam proses klasifikasi umumnya menggunakan rumus *euclidean distance* (Jayadi, B.V., et al. 2023). *Euclidean distance* adalah pengukuran jarak antara dua titik dalam ruang *Euclidean* (Wahyudi, A., & Utami, R., 2022). Sehingga penulis akan mencoba memodifikasi dengan menggunakan *normalized euclidean distance*.

Citra ialah suatu gambaran atau kemiripan dari suatu objek. Citra adalah hasil dari sistem perekaman data yang umumnya berupa foto, bersifat digital dan dapat langsung disimpan di media penyimpanan (Sunandar, H., 2017). Citra umumnya sering digunakan dalam proses klasifikasi buah. Dalam pengolahan klasifikasi citra, umumnya dibutuhkan beberapa ciri citra. Dalam tingkat proses mengekstraksi citra, merupakan penyampaian pada citra yang sangat penting untuk mengetahui hasil klasifikasi dari citra tersebut. Terdapat beberapa jenis ekstraksi dalam ekstraksi ciri

citra, salah satunya adalah warna. Untuk menilai kualitas buah jeruk yang baik, dapat dilakukan analisis berdasarkan warna yang diamati (Maulana, S., 2021).

Buah jeruk ialah buah yang termasuk dalam genus Citrus. Buah jeruk mengandung banyak vitamin C yang bermanfaat bagi kesehatan tubuh. Buah jeruk cenderung memiliki rasa asam, segar, dan manis. Buah jeruk biasanya memiliki warna orange, tetapi ada juga yang berwarna kuning dan hijau. Kandungan vitamin C dalam buah jeruk ialah 97,3 miligram atau setara dengan 163 % dari angka kecukupan gizi (AKG) (Kompas, 2023). jeruk termasuk juga tanaman buah yang telah dikenal dan ditanam sejak zaman dahulu dan berasal dari Asia. Pertanian buah jeruk di Indonesia mengalami kemajuan yang sangat cepat. Pada tahun 2019, luas lahan pertanian jeruk di Indonesia telah mencapai 73.083 ha. Daerah tempat pertanian jeruk juga telah menyebar hampir di semua Provinsi di Indonesia (Alfariza, E., & Chandra Artono, A., 2023). Dalam pemilihan buah jeruk yang memiliki kualitas baik, pemilihan buah jeruk dapat dilakukan secara manual atau dengan cara menebak buah yang dipilih layak untuk dikonsumsi. Saat memilih buah jeruk berkualitas baik, pengamatan secara manual dilakukan dengan memeriksa warna buah. Namun, metode pemilihan manual cenderung kurang efektif karena ada kemungkinan terjadinya kesalahan dalam pemilihan akibat keterbatasan indra penglihatan manusia.

Berdasarkan permasalahan tersebut, maka diperlukan solusi yang dapat membantu dalam pengklasifikasian kualitas jeruk lokal. Dalam tulisan ini, penulis akan memberikan gambaran dengan adanya suatu sistem pengolahan klasifikasi buah jeruk menggunakan metode *K-Nearest Neighbor* (KNN) yang dapat memberikan solusi dan

memecahkan masalah. Sehingga membantu penulis menetapkan kualitas buah jeruk lokal ke dalam kategori ‘kualitas baik’ dan ‘kualitas tidak baik’, yang penulis tuangkan dalam penelitian ini dengan mengambil topik untuk dijadikan bahan laporan skripsi dengan judul **“Modifikasi Metode K-Nearest Neighbor Dengan Normalized Euclidean Distance Untuk Klasifikasi Kualitas Jeruk Lokal Berastagi”**.

1.2 Rumusan Masalah

Berdasarkan penjelasan pada latar belakang di atas, berikut adalah rumusan masalah yang dapat dianalisis pada penelitian ini ialah, *K-Nearest Neighbor* (KNN) yang merupakan teknik dalam klasifikasi kualitas jeruk lokal berastagi berdasarkan warna buah, dan melihat bagaimana tingkat efektifitas metode *K-Nearest Neighbor* (KKN) untuk pengklasifikasian kualitas buah jeruk lokal berastagi.

1.3 Batasan Masalah

Berdasarkan penjelasan diatas, maka batasan masalah yang ditetapkan adalah:

1. Klasifikasi kualitas buah yang diteliti ialah buah jeruk lokal berastagi.
2. Nilai klasifikasi yang digunakan pada penelitian ini ialah warna buah (nilai RGB).
3. Pengambilan data buah jeruk menggunakan hasil kamera smartphone dan nikon.
4. Format gambar buah jeruk berupa jpg.
5. Metode yang diterapkan untuk menilai tingkat efektifitas kualitas buah ialah menggunakan metode *K-Nearest Neighbor* (KNN).

6. Sistem klasifikasi kualitas jeruk lokal berastagi menggunakan metode *K-Nearest Neighbor* (KNN) dibuat dengan menggunakan pemrograman Matlab.

1.4 Tujuan Penelitian

Dengan adanya latar belakang yang telah dijelaskan sebelumnya, tujuan penelitian ini adalah sebagai berikut:

1. Mengembangkan sistem pengklasifikasian kualitas jeruk lokal berastagi berdasarkan warna buah.
2. Menilai tingkat efektifitas metode *K-Nearest Neighbor* (KKN) dalam pengklasifikasian kualitas buah jeruk lokal berastagi.

1.5 Manfaat Penelitian

Penulisan skripsi ini diharapkan memberikan manfaat bagi semua pihak yang terkait antara lain:

1. Membantu untuk mengetahui kualitas buah jeruk yang baik berdasarkan warna buah.
2. Dengan adanya sistem klasifikasi kualitas buah jeruk ini, diharapkan dapat memberikan pengetahuan bagaimana cara kerja sistem klasifikasi dengan *K-Nearest Neighbor*.

BAB II

LANDASAN TEORI

2.1 Data Mining

Data mining merupakan proses penting di mana metode kecerdasan digunakan untuk mengekstrak pola data. *Data mining* melibatkan analisis observasional terhadap sekumpulan data untuk menemukan hubungan tidak terduga dan merangkum data dengan cara baru yang mudah dipahami oleh pemilik data (Novyanda, A., 2017).

Data mining adalah ilmu yang digunakan untuk mengungkap pengetahuan yang terdapat dalam database. Proses ini memanfaatkan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk menghasilkan dan menemukan informasi yang berguna dari berbagai database (Utomo, D.P., & Mesran, 2020). Tujuan utama *data mining* ialah untuk menemuka atau menggali, atau menambah pengetahuan dari data dan informasi yang dimiliki (Rahayuningsih, P.A., 2019).

2.2 Klasifikasi

Klasifikasi adalah salah satu algoritma dalam *data mining* yang bertujuan untuk memungkinkan komputer atau prosesor melakukan pembelajaran dengan metode dan algoritma yang sesuai dengan konsep machine learning. Klasifikasi melibatkan data pelatihan dan pengujian dalam bentuk teks, gambar, dan video (Al-Khowarizmi & Suherman, 2021).

Klasifikasi ialah sebuah cara untuk mengelompokkan suatu objek berdasarkan ciri yang dimiliki objek yang ingin di klasifikasi. Klasifikasi menggunakan teknik yang

memperhatikan perilaku dan atribut dari kelompok yang telah didefinisikan. Teknik ini dapat mengklasifikasikan data baru dengan memanipulasi data yang sudah diklasifikasikan sebelumnya dan menggunakan hasil tersebut untuk menghasilkan sejumlah aturan (Fitriani, E., 2020).

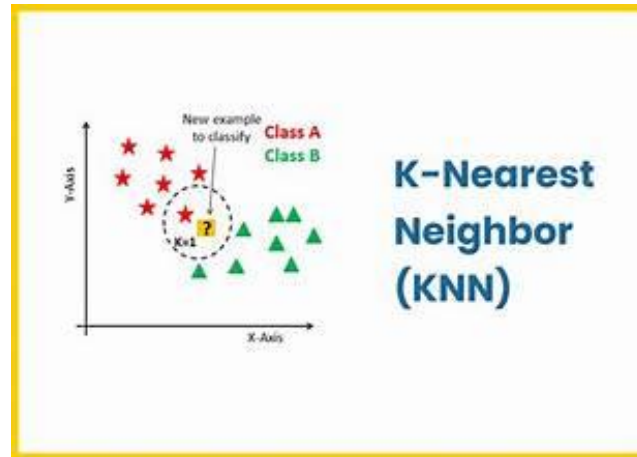
Klasifikasi adalah proses yang bertujuan untuk mengidentifikasi kumpulan pola atau fungsi yang mendeskripsikan dan memisahkan satu kelas dari kelas lainnya, sehingga objek tersebut dapat dikategorikan ke dalam kategori tertentu yang telah ditetapkan. Klasifikasi merupakan bentuk analisa yang menghasilkan model yang menggambarkan kelas data (Novandya, A., 2017).

2.3 Algoritma *K-Nearest Neighbor* (KNN)

Algoritma KNN adalah algoritma non numerik yang dikenal dalam data mining, yang dapat digunakan untuk melakukan klasifikasi maupun regresi. Untuk melakukan klasifikasi menggunakan sebuah algoritma tertentu membutuhkan sebuah dataset yang terdiri dari data training dan data testing. Algoritma KNN merupakan sebuah proses supervised yang memiliki arti bahwa pada proses KNN ini memerlukan informasi training untuk dilakukannya klasifikasi terhadap suatu objek yang memiliki jarak terdekat. Adapun prinsip kerja pada KNN ini adalah dengan mencari jarak euclidian atau jarak terdekat berdasarkan nilai k (A'Yunyah, Q., & Reza, M., 2023).

Algoritma KNN adalah sebuah metode yang digunakan untuk mengklasifikasikan objek berdasarkan data pembelajaran yang memiliki jarak terdekat dengan objek tersebut. Tujuan dari algoritma ini adalah untuk mengklasifikasikan objek baru berdasarkan atribut dari *training sample* (Yustanti, W., 2012). Berdasarkan penjelasan

diatas, dapat disimpulkan bahwa metode KNN adalah suatu metode yang dapat mengklasifikasikan suatu objek berdasarkan data terdekat dari tetangga atau data sebelumnya yang dimiliki sebagai sample untuk mencapai hasil akhir.



Gambar 2.1 Algoritma *K-Nearest Neighbor* (KNN)

(sumber: www.trivusi.web.id)

2.4 Jeruk Lokal

Jeruk lokal ialah buah yang banyak dikonsumsi oleh masyarakat Berastagi. Buah ini kaya akan vitamin C, vitamin B6, antioksidan dan serat. Oleh karena itu, sangat disarankan untuk mengonsumsi jeruk setiap hari, karena dapat membantu melancarkan pencernaan, mengurangi risiko diabetes, menjaga kesehatan kulit, serta meningkatkan daya tahan tubuh. Jeruk lokal memiliki kandungan vitamin C yang tinggi, yang sangat penting bagi kesehatan tubuh (Pawening, R.E., et al. 2020).

Jeruk juga bermanfaat untuk pertumbuhan tulang pada anak – anak karena selain mengandung vitamin C, jeruk juga mengandung zat fosfor dan zat kapur. Kandungan vitamin C dalam jeruk dapat membantu mencegah penyakit influenza. Buah jeruk

memiliki potensi yang sangat baik untuk dikembangkan karena dapat ditemukan disetiap musim. Selain itu, jeruk juga sangat mudah untuk ditanam karena dapat bertahan di semua cuaca, baik dataran rendah maupun dataran tinggi (Siwilopo, K.P., & Marcos, H., 2023).



Gambar 2.2 Jeruk Lokal Berastagi

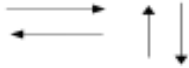





2.5 Matlab

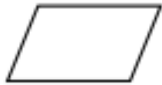





Matlab (*Matrix Laboratory*) adalah sebuah program yang digunakan untuk analisis dan komputasi numerik, serta merupakan bahasa pemrograman matematika tingkat lanjut yang didasarkan pada sifat dan bentuk matriks. Pada awalnya, program ini berfungsi sebagai antarmuka untuk koleksi rutin numeric dari proyek LINPACK dan EISPACK, yang dikembangkan menggunakan bahasa FORTRAN. Saat ini, Matlab menjadi produk komersial dari perusahaan Mathworks, Inc dan dalam perkembangan selanjutnya menggunakan bahasa C++ dan assembler (terutama untuk fungsi-fungsi dasar Matlab) (Putra, R.T., 2023).

2.6 Flowchart

Flowchart adalah metode untuk menuliskan algoritma menggunakan notasi grafis. Flowchart merupakan diagram yang menunjukkan urutan atau langkah-langkah dalam suatu program serta hubungan antar proses beserta pernyataannya. Setiap simbol dalam diagram ini menggambarkan proses tertentu, sedangkan hubungan antar proses ditunjukkan dengan garis penghubung (Fauzi, J.R., 2020). Simbol – simbol dalam flowchart:

Tabel 2.1 Simbol *Flowchart*

Simbol	Nama	Fungsi
	Flow	Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain
	On-page	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang sama
	Off-page	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang berbeda
	Terminator	Simbol yang menyatakan awal atau akhir suatu program
	Process	Simbol yang menyatakan suatu proses yang dilakukan komputer
	Decision	Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban yaitu ya dan tidak

	Input/Output	Simbol yang menyatakan proses input atau output tanpa tergantung peralatan
	Manual Operation	Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer
	Document	Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik atau output yang perlu dicetak
	Predefine Process	Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur
	Display	Simbol yang menyatakan peralatan output yang digunakan
	Preperation	Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal





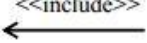
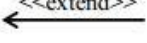
Sumber: (www.dicoding.com)

2.7 UML

UML (Unified Modeling Language) adalah bahasa yang paling umum digunakan untuk spesifikasi kebutuhan. UML adalah bahasa pemodelan grafis standar yang mencakup serangkaian diagram yang terintegrasi. Setiap diagram menggambarkan sistem pemodelan dalam bagian yang berbeda, tetapi secara keseluruhan diagram tersebut dapat memberikan peta lengkap dari sistem pemodelan (Elsayed, M., et al. 2020). Jenis – jenis dalam membuat suatu diagram UML, yaitu:

a. *Use Case Diagram*

Use Case Diagram ialah suatu teknik yang dibuat untuk menggambarkan proses aktivitas sistem (Sari, I.P., et al. 2022). *Use Case Diagram* menggambarkan fungsionalitas sistem, dengan penekanan pada apa yang dilakukan sistem, bukan bagaimana cara melakukannya. *Use Case Diagram* dibuat untuk menunjukkan interaksi antara actor dan sistem (Sari, I.P., et al. 2022).

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi






Gambar 2.3 Simbol *Use Case Diagram*

(sumber: badoystudio.com)

b. Activity Diagram

Activity Diagram adalah diagram yang menggambarkan alur kerja atau aktivitas dalam program yang sedang dikembangkan (Sari, I.P., et al. 2022).

Diagram ini dibuat bertujuan agar alur kerja lebih terstruktur. Berikut simbol – simbol Activity Diagram:

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir

Gambar 2.4 Simbol Activity Diagram

(Sumber: imagesee.biz)

c. Sequence Diagram

Pada tahap ini Sequence Diagram menjelaskan secara detail urutan proses yang dilakukan sistem untuk mencapai tujuan dari Use Case. Interaksi yang terjadi antar class, operasi apa saja yang terlibat, urutan antar operasi, dan informasi

yang dibutuhkan oleh setiap operasi (Sari, I.P., et al. 2022). Berikut simbol – simbol dari Sequence Diagram:

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.
2		<i>Life Line</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
3		<i>Object Message</i>	Menggambarkan pesan/hubungan antar objek yang menunjukkan urutan kejadian yang terjadi.
4		<i>Message to Self</i>	Menggambarkan pesan/hubungan objek itu sendiri yang menunjukkan urutan kejadian yang terjadi.

Gambar 2.5 Simbol Sequence Diagram

(Sumber: widuri.raharja.info)

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian


Metode pengumpulan data yang digunakan dalam penelitian ini meliputi observasi dan studi pustaka. Observasi dilakukan secara langsung di pasar buah Berastagi. Sementara studi pustaka dilakukan dengan mengumpulkan data dan informasi dari buku, jurnal, serta artikel yang relevan dengan klasifikasi.





Metode observasi adalah teknik pengumpulan data yang dilakukan melalui pengamatan dan pencatatan fenomena-fenomena yang sedang diteliti secara sistematis, observasi ini mencakup pengamatan yang dilakukan baik secara langsung maupun tidak langsung (Hasibuan, M.P., et al. 2023).




3.2 Dataset

Pada penelitian ini, penulis memiliki dataset berjumlah 100 gambar jeruk, dimana 100 gambar tersebut terbagi menjadi 2 yaitu jeruk baik dan jeruk buruk.

Tabel 3.1 Dataset

NO	KETERANGAN	GAMBAR
1	Jeruk Baik	

2	Jeruk Baik	
3	Jeruk Buruk	
4	Jeruk Buruk	
5	Jeruk Baik	
..
..

98	Jeruk Baik	
99	Jeruk Buruk	
100	Jeruk Baik	

3.3 Normalized

Normalized adalah proses penyesuaian skala fitur atau variabel input sehingga semuanya berada dalam rentang nilai yang sama, biasanya antara 0 dan 1. Dataset yang digunakan adalah gambar buah jeruk. Metode normalisasi yang diterapkan dalam penelitian ini adalah:

- Min-max Normalized

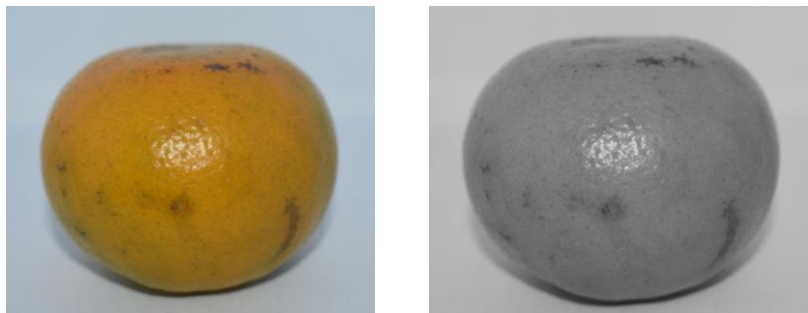
Min-max normalized merupakan metode normalisasi yang melakukan transformasi linier pada data asli, sehingga menghasilkan keseimbangan nilai perbandingan antar data sebelum dan sesudah proses (Nasution, D.A, et al.2019). Metode ini dapat diterapkan dengan menggunakan rumus:

$$N = \frac{n - X_{min}}{X_{max} - X_{min}} \dots\dots\dots (3.1)$$

n = nilai asli

Xmin = nilai terendah

Xmax = nilai tertinggi

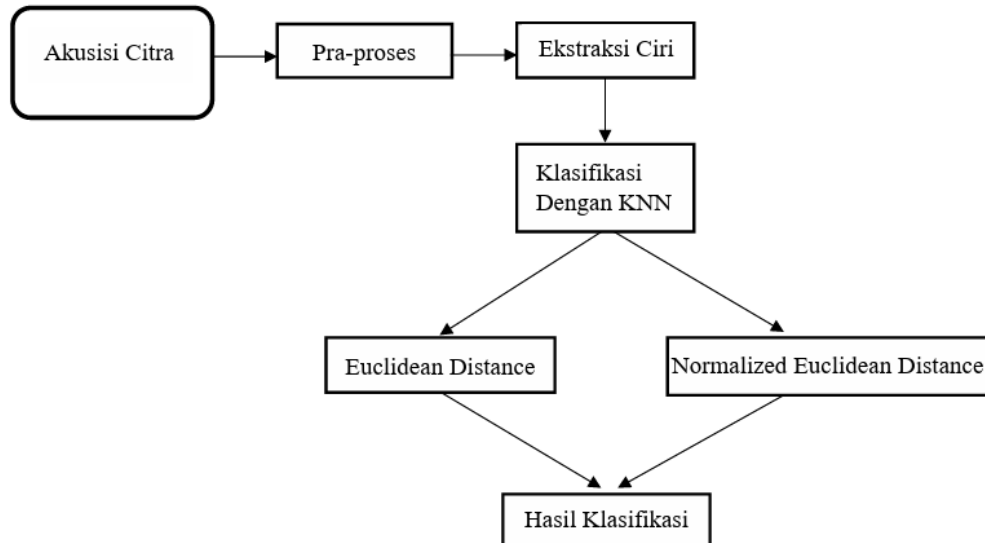


Gambar 3.1 Contoh Normalisasi Buah Jeruk

3.4 Prosedur Penelitian

Penelitian ini terdiri dari beberapa tahap, yaitu akuisisi citra, praproses, ekstraksi ciri, dan klasifikasi. Akuisisi citra dilakukan dengan menggunakan kamera *smartphone* dan kamera nikon. Tahap praproses mencakup perubahan ukuran citra. Ekstraksi ciri

yang digunakan adalah warna buah. Sedangkan klasifikasi citra menerapkan algoritma *K-Nearest Neighbor*.



Gambar 3.2 Proses Tahapan Penelitian

a. Akusisi citra

Tahap awal sebelum pemrosesan, tahap ini merupakan proses pengambilan citra menggunakan kamera smartphone dan kamera nikon.

b. Pra-proses

Pra-proses ialah tahap pengolahan citra agar mendapatkan kualitas citra yang baik. Beberapa tahapan dalam pra-proses ialah resize dan cropping untuk memperkecil ukuran gambar.

c. Ekstraksi ciri

Pada tahapan ini ekstraksi ciri citra yang digunakan ialah warna citra, dengan mengambil nilai RGB (*Red, Green, dan Blue*) untuk menghitung jarak *Euclidean*.

d. Klasifikasi dengan KNN

Klasifikasi menggunakan metode KNN dengan rumus *euclidean distance*

$$D(a, b) = \sqrt{\sum_{k=1}^d (a_k - b_k)^2} \dots\dots\dots (3.2)$$

Klasifikasi menggunakan rumus modifikasi *normalized euclidean distance*

$$D_{norm}(a, b) = \sqrt{\sum_{k=1}^d \left(\frac{a_k - b_k}{range_k} \right)^2} \dots\dots\dots (3.3)$$

$D(a,b)$: jarak skalar antara dua vektor data a dan b

D : ukuran dimensi

k : jumlah tetangga terdekat

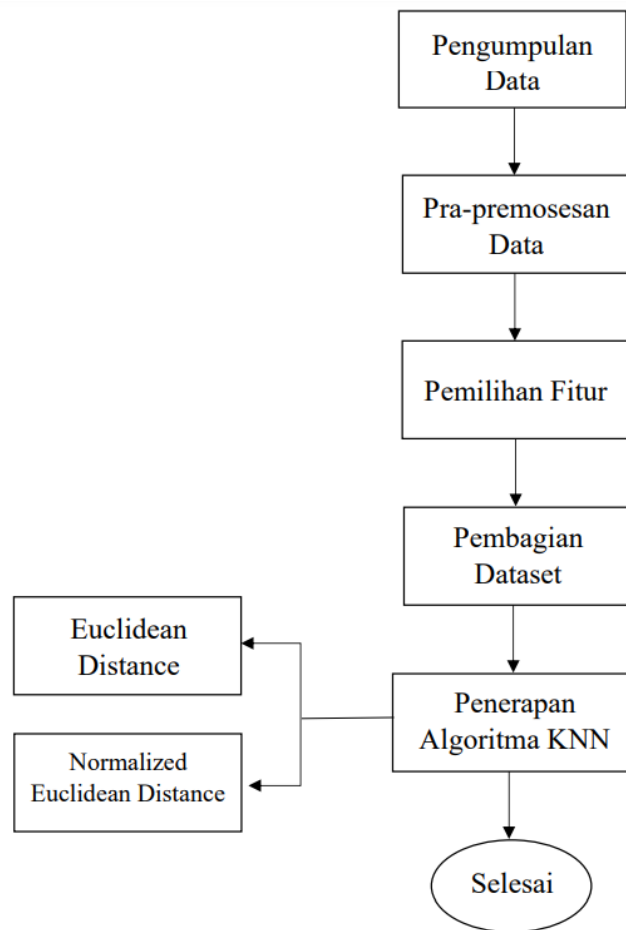
Range k : rentang (nilai maksimum – nilai minimum) dari dimensi ke- i dalam dataset.

e. Hasil klasifikasi

Dari hasil klasifikasi, dapat disimpulkan apakah metode K-Nearest Neighbor (KNN) dalam penelitian ini mampu mengklasifikasikan citra jeruk sebagai kualitas “baik” atau “tidak baik”.

3.5 Alur Perancangan Sistem

Penelitian ini memiliki beberapa alur yaitu, pengumpulan data, pra-premosesan data, pemilihan fitur, pembagian dataset, dan penerapatan algoritma KNN.



Gambar 3.3 Alur Perancangan Sistem

a. Pengumpulan Data

Pengumpulan data dilakukan untuk mengumpulkan dataset jeruk lokal Berastagi.

b. Pra-premosesan Data

Pada tahap ini, dilakukan untuk mempersiapkan dataset agar lebih baik dan akurat.

c. Pemilihan Fitur

Pemilihan fitur yang berkaitan dengan klasifikasi buah jeruk seperti fitur warna.

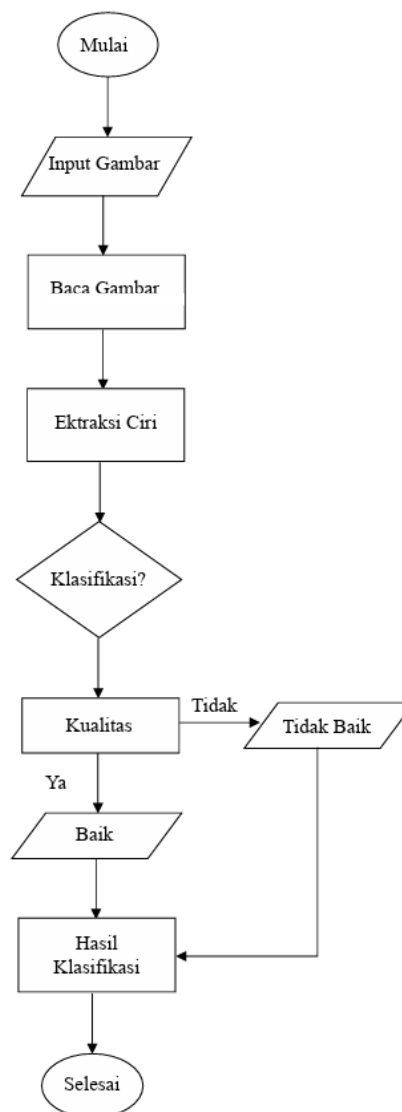
d. Pembagian Dataset

Dilakukan pembagian data menjadi dataset pelatihan dan pengujian.

e. Penerapan Algoritma KNN,

Dilakukan dengan dua cara yaitu, *Euclidean Distance* dan *Normalized Euclidean Distance*.

3.6 Flowchart Sistem Klasifikasi



Gambar 3.4 Flowchart Sistem Klasifikasi

Pada gambar diatas, flowchart sistem klasifikasi menggambarkan alur kerja sistem untuk mengetahui kualitas jeruk yang di input ke dalam sistem sesuai dengan data ciri gambar jeruk yang terdapat pada sistem. Adapun sistematika flowchart dibawah yaitu, diawali dengan menginput gambar jeruk. Setelah gambar diinput, kemudian sistem akan membaca gambar. Setelah itu, sistem akan mengekstraksi ciri gambar yang telah di input, kemudian sistem akan menentukan klasifikasi kualitas jeruk, jika jeruk tersebut memiliki ciri yang baik maka kualitas jeruk baik, dan sebaliknya jika jeruk memiliki ciri yang buruk maka kualitas jeruk tidak baik.

3.7 Unified Modeling Language (UML) Diagram



Gambar 3.5 UML Diagram

Adapun UML Diagram pada penelitian ini terdapat 4 bagian yaitu, kelas GUI, kelas image processor, kelas KNN Classifier, Kelas Data Set.

a. Kelas GUI

Atribut:

- 1) loadImageButton: Tombol untuk memuat gambar dari file.
- 2) cropImageButton: Tombol untuk memotong gambar.
- 3) segmentImageButton: Tombol untuk melakukan segmentasi gambar.
- 4) extractFeatureButton: Tombol untuk mengekstrak fitur dari gambar.
- 5) knnButton: Tombol untuk menjalankan klasifikasi KNN.
- 6) euclideanDistanceRadioButton: Tombol radio untuk memilih metrik jarak Euclidean.
- 7) normalizedEuclideanDistanceRadioButton: Tombol radio untuk memilih metrik jarak Euclidean yang dinormalisasi.
- 8) imageSample: Area untuk menampilkan sampel gambar.
- 9) croppedImage: Area untuk menampilkan gambar yang telah dipotong.
- 10) segmentedImage: Area untuk menampilkan gambar yang telah disegmentasi.
- 11) testImage: Area untuk menampilkan gambar uji.
- 12) featureRed, featureGreen, featureBlue: Kolom untuk menampilkan nilai fitur warna.
- 13) trainDataGood, trainDataBad: Kolom untuk menampilkan data latih untuk klasifikasi "baik" dan "buruk".
- 14) classificationResult: Kolom untuk menampilkan hasil klasifikasi.

Metode:

- 1) `loadImage()`: Memuat gambar dari file.
- 2) `cropImage()`: Memotong gambar yang telah dimuat.
- 3) `segmentImage()`: Melakukan segmentasi pada gambar.
- 4) `extractFeatures()`: Mengekstrak fitur dari gambar.
- 5) `classifyKNN()`: Menjalankan klasifikasi KNN pada gambar yang telah diproses.

b. Kelas Image Processor

Metode:

- 1) `loadImage(filePath)`: Memuat gambar dari jalur file yang diberikan.
- 2) `cropImage(image)`: Memotong bagian dari gambar sesuai dengan koordinat yang ditentukan.
- 3) `segmentImage(image)`: Melakukan segmentasi pada gambar untuk memisahkan objek dari latar belakang.
- 4) `extractFeatures(image)`: Mengekstrak fitur dari gambar seperti warna (misalnya, nilai rata-rata untuk merah, hijau, dan biru).

c. Kelas KNN Classifier

Atribut:

- 1) `k`: Jumlah tetangga terdekat yang akan diperhitungkan dalam proses klasifikasi.
- 2) `distanceMetric`: Metrik jarak yang digunakan (misalnya, Euclidean atau Normalized Euclidean).

Metode:

- 1) `train(trainingData)`: Melatih model KNN dengan data pelatihan yang diberikan.

- 2) `classify(testData)`: Mengklasifikasikan data uji berdasarkan model yang telah dilatih.
- 3) `computeEuclideanDistance()`: Menghitung jarak Euclidean antara dua titik data.
- 4) `computeNormalizedEuclideanDistance()`: Menghitung jarak Euclidean yang dinormalisasi antara dua titik data.

d. Kelas Data Set

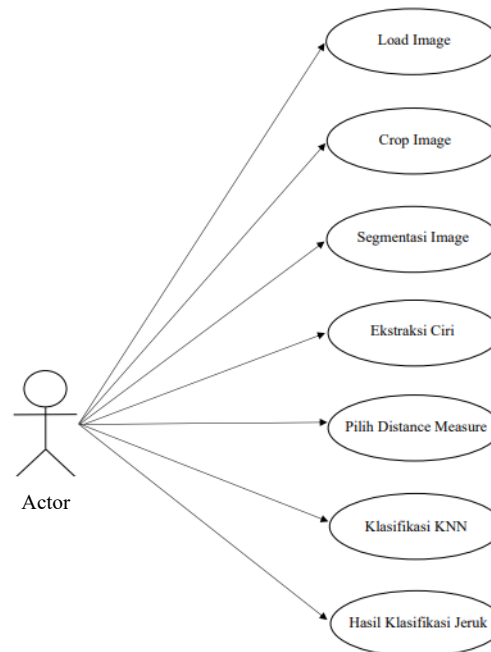
Atribut:

- 1) `trainingData`: Kumpulan data yang digunakan untuk melatih model.
- 2) `testingData`: Kumpulan data yang digunakan untuk menguji kinerja model.

Metode:

- 1) `splitData(data)`: Membagi kumpulan data menjadi dua bagian.

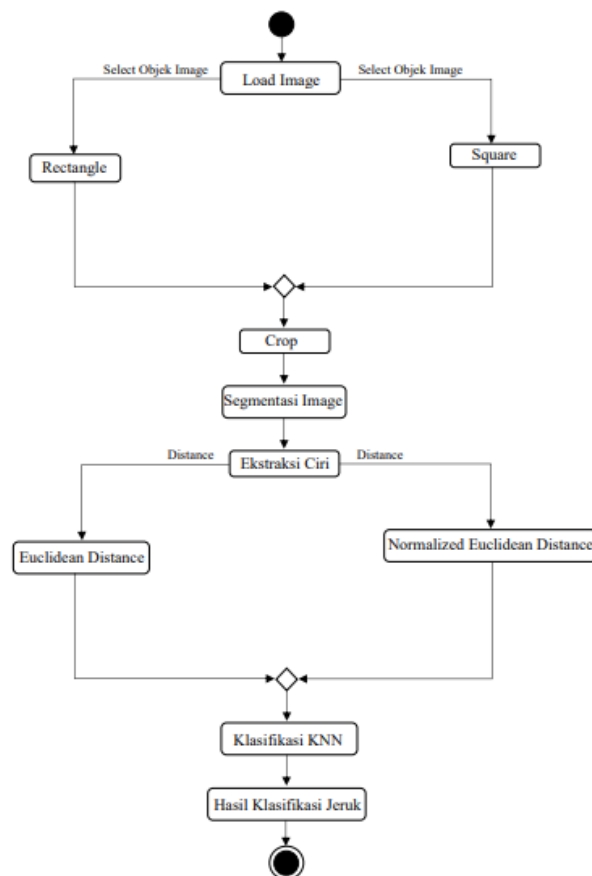
3.8 Use Case Diagram



Gambar 3.6 Use Case Diagram

Actor pada sistem ini ialah pengguna. Untuk pengguna memiliki beberapa kegiatan yang mereka lakukan dalam sistem klasifikasi kualitas jeruk Berastagi. Pertama, pengguna dapat memasukkan gambar ke dalam sistem. Pengguna juga dapat melakukan cropping gambar pada menu crop dalam sistem. Setelah gambar dicrop, pengguna dapat memilih menu segmentasi agar sistem melakukan proses segmentasi. Setelah proses segmentasi selesai, pengguna dapat melihat nilai RGB gambar pada menu ekstraksi citra. Pada menu klasifikasi KNN, pengguna memiliki opsi untuk proses pengukuran jarak dengan Euclidean Distance atau Normalized Euclidean Distance. Setelah melakukan klasifikasi KNN maka hasil klasifikasi akan keluar.

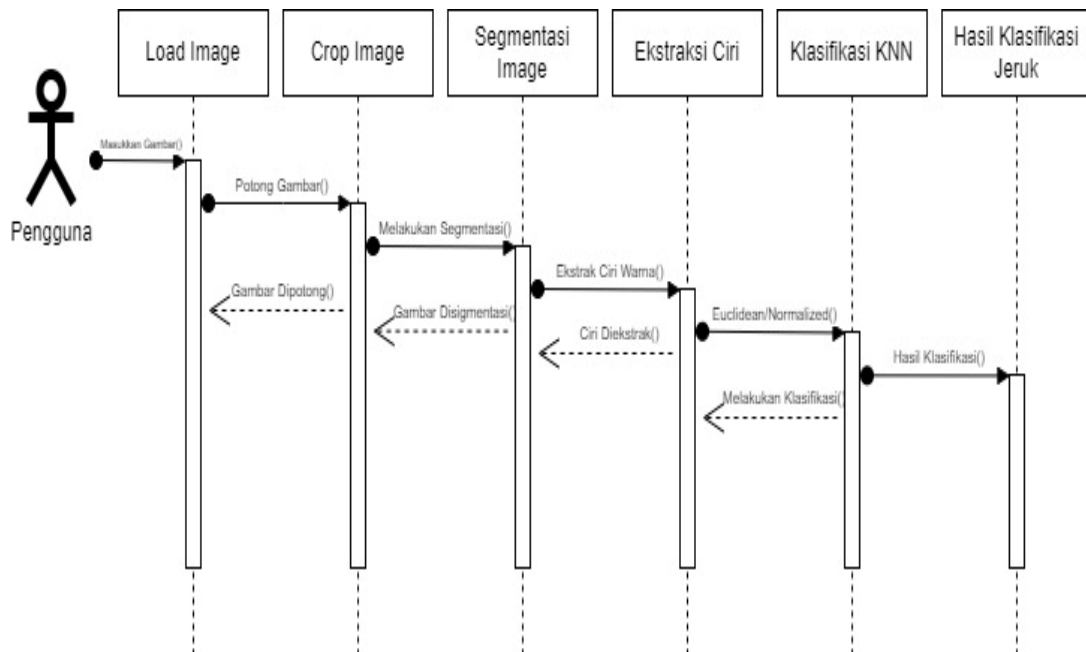
3.9 Activity Diagram



Gambar 3.7 Activity Diagram

Pada gambar diatas, pengguna memulai dengan menu load gambar yang ingin di-klasifikasikan, lalu apabila ingin melakukan cropping maka pengguna memiliki dua opsi yaitu, rectangle dan square. Selanjutnya pengguna memilih segmentasi image lalu sistem akan mengeluarkan hasil segmentasi. Pada menu ekstraksi ciri, pengguna dapat melihat ciri warna gambar yang ingin diklasifikasi. Lalu pada klasifikasi KNN, pengguna juga memiliki dua opsi yang dapat dipilih. Setelah pengguna melakukan proses klasifikasi KNN maka hasil klasifikasi jeruk akan keluar.

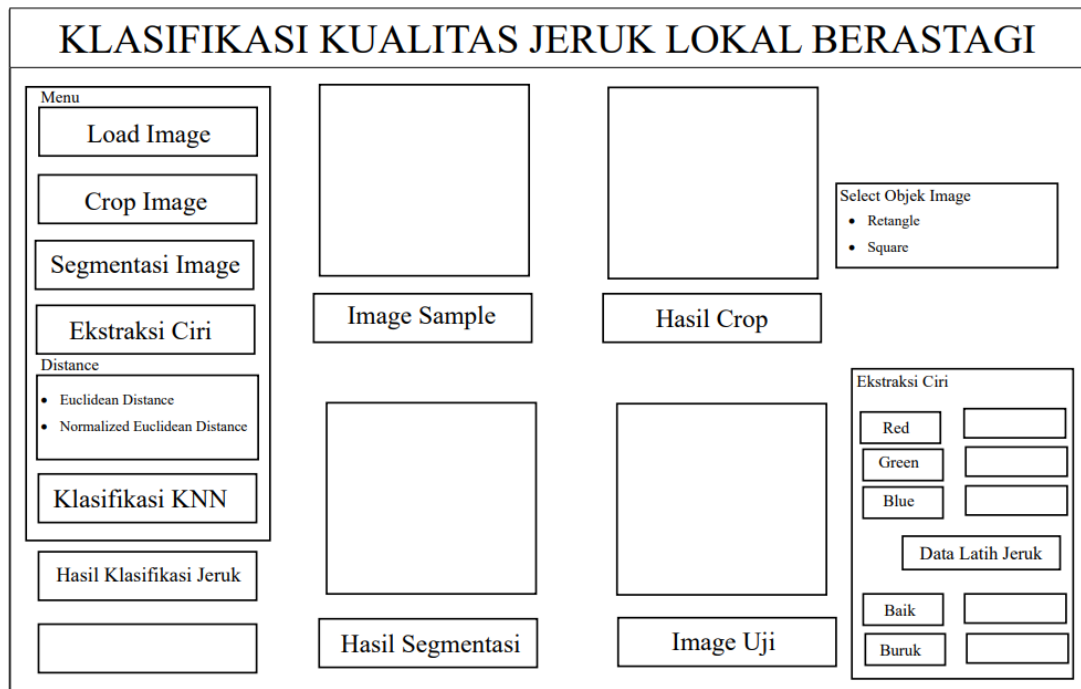
3.10 Sequence Diagram



Gambar 3.8 Sequence Diagram

Pada gambar diatas, pengguna memasukkan gambar, lalu pada menu crop image, pengguna memotong gambar, gambar dipotong. Segmentasi image, melakukan segmentasi lalu gambar disegmentasi. Ekstraksi ciri, sistem mengekstrak ciri warna. Klasifikasi KNN terdapat 2 opsi. Setelah itu maka keluar hasil klasifikasi jeruk.

3.11 Antar Muka Sistem



Gambar 3.9 Tampilan Antar Muka Sistem

Adapun tampilan antar muka sistem yang dikembangkan dalam penelitian ini dapat dilihat pada gambar diatas ini. Pada tampilan terdapat nama sistem yaitu “Klasifikasi Kualitas Jeruk Lokal Berastagi”. Terdapat beberapa menu seperti load image, crop image, segmentasi image, ekstraksi ciri, klasifikasi KNN, dan hasil klasifikasi. Terdapat juga tampilan image sample, hasil crop, hasil segmentasi, dan image uji. Pada kolom ekstraksi ciri, terdapat nilai RGB gambar yang sedang diuji dan jumlah data latih.

BAB IV

HASIL DAN PEMBAHASAN

Pada penelitian ini, klasifikasi kualitas jeruk lokal Berastagi dilakukan dengan metode KNN (*K-Nearest Neighbor*) dengan Normalized Euclidean Distance. Fitur yang digunakan meliputi nilai warna R, G, dan B. Setelah identifikasi selesai, evaluasi hasil dilakukan dengan mengukur tingkat kualitas jeruk lokal Berastagi dari sistem yang dikembangkan.

4.1 Tampilan Sistem



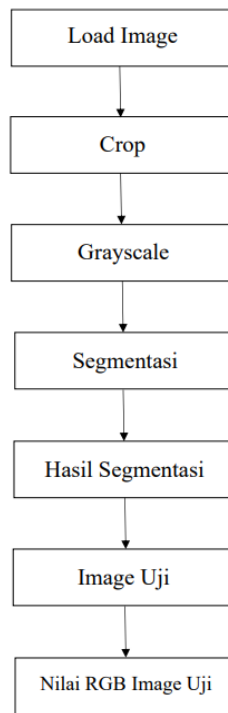
Gambar 4.1 Tampilan Sistem

Pembuatan sistem dilakukan untuk mengklasifikasi kualitas jeruk lokal berastagi dengan menggunakan Matlab. Pada sistem akan menampilkan kolom untuk menginput sampel gambar yang akan diuji, kemudian pada menu crop image user dapat meng-crop sampel gambar, terdapat 2 pilihan yaitu rectangle dan square. Pada menu segmentasi image akan keluar hasil segmentasi, menu ekstraksi ciri akan keluar hasil

nilai RGB. Lalu pada menu klasifikasi KNN dapat dilakukan dengan *Euclidean Distance* dan *Normalized Euclidean Distance* dan user dapat melihat hasil klasifikasi sampel jeruk.

4.2 Nilai RGB (*Red, Green, Blue*)

Ada pun proses mendapatkan nilai RGB menggunakan Matlab perintah konding sebagai berikut:



Gambar 4.2 Proses Nilai RGB

Pengguna akan meng-input dan meng-crop gambar yang akan diuji, sistem akan membaca gambar lalu sistem akan merubah gambar menjadi grayscale. Lalu sistem melakukan proses segmentasi sampai hasil segmentasi. Ketika hasil segmentasi sudah keluar maka sistem akan memproses hasil segmentasi menjadi image uji dan nilai RGB dari image uji tersebut seperti pada gambar di bawah ini.



Gambar 4.3 Contoh Pengujian Sampel 3

Untuk 19 sampel lainnya dapat dilihat pada tabel di bawah ini:

Tabel 4.1 Nilai RGB Sampel Jeruk

NO	DATA JERUK	RED	GREEN	BLUE	KUALITAS
1	Sampel 1	158.7066	131.8587	80.445	BAIK
2	Sampel 2	166.4275	138.94	97.3779	BURUK
3	Sampel 4	164.4274	130.3507	73.7947	BAIK
4	Sampel 5	155.8918	126.3104	64.4122	BAIK
5	Sampel 6	185.1111	138.6269	84.5178	BURUK
6	Sampel 7	137.2475	104.47	64.4259	BURUK
7	Sampel 8	134.1979	101.6341	64.4494	BURUK
8	Sampel 9	173.6608	136.2476	75.5207	BAIK
9	Sampel 10	122.9824	88.7374	55.6396	BURUK
10	Sampel 11	194.9772	146.1809	85.2512	BAIK
11	Sampel 12	152.5341	121.4093	72.609	BAIK
12	Sampel 13	158.397	138.8766	73.1123	BURUK
13	Sampel 14	165.9128	140.7492	83.4876	BAIK
14	Sampel 15	161.864	142.511	86.3462	BURUK
15	Sampel 16	133.0953	130.2986	90.4749	BURUK
16	Sampel 17	177.4837	137.2362	80.6419	BAIK
17	Sampel 18	173.557	137.1082	77.9342	BAIK
18	Sampel 19	138.7763	127.7308	81.5836	BURUK
19	Sampel 20	129.1697	93.7282	48.5259	BURUK

4.3 Euclidean Distance

Tabel 4.2 Sampel Yang Akan Diuji

NO	DATA JERUK	RED	GREEN	BLUE	KUALITAS
1	Sampel 1	158.7066	131.8587	80.445	BAIK
2	Sampel 2	166.4275	138.94	97.3779	BURUK
3	Sampel 3	182.2807	153.9614	96.3495	BAIK
4	Sampel 4	164.4274	130.3507	73.7947	BAIK
5	Sampel 5	155.8918	126.3104	64.4122	BAIK
6	Sampel 6	185.1111	138.6269	84.5178	BURUK
7	Sampel 7	137.2475	104.47	64.4259	BURUK
8	Sampel 8	134.1979	101.6341	64.4494	BURUK
9	Sampel 9	173.6608	136.2476	75.5207	BAIK
10	Sampel 10	122.9824	88.7374	55.6396	BURUK
11	Sampel 11	194.9772	146.1809	85.2512	BAIK
12	Sampel 12	152.5341	121.4093	72.609	BAIK
13	Sampel 13	158.397	138.8766	73.1123	BURUK
14	Sampel 14	165.9128	140.7492	83.4876	BAIK
15	Sampel 15	161.864	142.511	86.3462	BURUK
16	Sampel 16	133.0953	130.2986	90.4749	BURUK
17	Sampel 17	177.4837	137.2362	80.6419	BAIK
18	Sampel 18	173.557	137.1082	77.9342	BAIK
19	Sampel 19	138.7763	127.7308	81.5836	BURUK
20	Sampel 20	129.1697	93.7282	48.5259	BURUK
21	Sampel 21	154.6181	148.8581	88.9745	?

Menghitung nilai *Euclidean Distance* digunakan untuk mengidentifikasi jeruk-jeruk yang paling mirip dengan jeruk yang sedang diuji. KNN akan menghitung jarak antara jeruk yang belum diklasifikasikan dengan setiap jeruk dalam dataset yang sudah diklasifikasikan, kemudian memilih k jeruk terdekat untuk menentukan kualitas jeruk yang sedang diuji tersebut berdasarkan mayoritas kualitas dari k tetangga terdekat. Penulis akan mencari nilai *Euclidean Distance* sampel jeruk dengan $k = 3$ menggunakan **Persamaan 3.2** sebagai berikut:

$$D(a,b,c) = \sqrt{\sum_{k=1}^d (a_k - b_k - c_k)^2}$$

$$D(a,b,c) = \sqrt{(158.7066 - 154.6181)^2 + (131.8587 - 148.8581)^2 + (80.445 - 88.9745)^2}$$

$$D(a,b,c) = \sqrt{(4.0885)^2 + (-16.9994)^2 + (-8.5295)^2}$$

$$D(a,b,c) = \sqrt{(16.71583225) + (288.97960036) + (72.75237025)}$$

$$D(a,b,c) = \sqrt{378.44780286}$$

$$D(a,b,c) = 19.4537349334$$

Kemudian setelah nilai *Euclidean Distance* dihitung maka nilai tersebut akan diurutkan berdasarkan nilai yang paling terkecil. Untuk nilai *Euclidean Distance* 20 sampel lainnya dapat dilihat pada tabel di bawah ini:

Tabel 4.3 Euclidean Distance

NO	DATA JERUK	RED	GREEN	BLUE	EUCLIDEAN DISTANCE	URUTAN	KLASIFIKASI
1	Sampel 1	158.7066	131.8587	80.445	19.45373493	5	BAIK
2	Sampel 2	166.4275	138.94	97.3779	17.5626811	3	BURUK
3	Sampel 3	182.2807	153.9614	96.3495	29.08012611	12	BAIK
4	Sampel 4	164.4274	130.3507	73.7947	25.86836967	7	BAIK
5	Sampel 5	155.8918	126.3104	64.4122	33.36656513	15	BAIK
6	Sampel 6	185.1111	138.6269	84.5178	32.47095128	14	BURUK
7	Sampel 7	137.2475	104.47	64.4259	53.61599508	17	BURUK
8	Sampel 8	134.1979	101.6341	64.4494	56.99623912	18	BURUK
9	Sampel 9	173.6608	136.2476	75.5207	26.50761906	8	BAIK
10	Sampel 10	122.9824	88.7374	55.6396	75.67385044	20	BURUK
11	Sampel 11	194.9772	146.1809	85.2512	40.61880495	16	BAIK
12	Sampel 12	152.5341	121.4093	72.609	32.02513494	13	BAIK
13	Sampel 13	158.397	138.8766	73.1123	19.11857255	4	BURUK

14	Sampel 14	165.9128	140.7492	83.4876	14.94759442	2	BAIK
15	Sampel 15	161.864	142.511	86.3462	9.98482379	1	BURUK
16	Sampel 16	133.0953	130.2986	90.4749	28.45939494	11	BURUK
17	Sampel 17	177.4837	137.2362	80.6419	26.96917584	9	BAIK
18	Sampel 18	173.557	137.1082	77.9342	24.8722799	6	BAIK
19	Sampel 19	138.7763	127.7308	81.5836	27.42172196	10	BURUK
20	Sampel 20	129.1697	93.7282	48.5259	72.95900341	19	BURUK
21	Sampel 21	154.6181	148.8581	88.9745	0		?

K = 3 BURUK

4.4 Normalized Euclidean Distance

Tabel 4.4 Nilai Yang Akan Diubah

NO	DATA JERUK	RED	GREEN	BLUE	KUALITAS
1	Sampel 1	158.7066	131.8587	80.445	BAIK
2	Sampel 2	166.4275	138.94	97.3779	BURUK
3	Sampel 3	182.2807	153.9614	96.3495	BAIK
4	Sampel 4	164.4274	130.3507	73.7947	BAIK
5	Sampel 5	155.8918	126.3104	64.4122	BAIK
6	Sampel 6	185.1111	138.6269	84.5178	BURUK
7	Sampel 7	137.2475	104.47	64.4259	BURUK
8	Sampel 8	134.1979	101.6341	64.4494	BURUK
9	Sampel 9	173.6608	136.2476	75.5207	BAIK
10	Sampel 10	122.9824	88.7374	55.6396	BURUK
11	Sampel 11	194.9772	146.1809	85.2512	BAIK
12	Sampel 12	152.5341	121.4093	72.609	BAIK
13	Sampel 13	158.397	138.8766	73.1123	BURUK
14	Sampel 14	165.9128	140.7492	83.4876	BAIK
15	Sampel 15	161.864	142.511	86.3462	BURUK
16	Sampel 16	133.0953	130.2986	90.4749	BURUK
17	Sampel 17	177.4837	137.2362	80.6419	BAIK
18	Sampel 18	173.557	137.1082	77.9342	BAIK
19	Sampel 19	138.7763	127.7308	81.5836	BURUK
20	Sampel 20	129.1697	93.7282	48.5259	BURUK
21	Sampel 21	154.6181	148.8581	88.9745	?

Tujuan *Normalized* ialah untuk mengubah nilai kolom numerik dalam kumpulan data ke dalam skala, tanpa mengubah perbedaan dalam rentang nilai. *Normalized* juga merupakan salah satu tahap yang dapat meningkatkan akurasi pada data. *Normalized* yang digunakan ialah Min-Max dengan rentang 0-1. Penulis terlebih dulu akan merubah nilai *Euclidean Distance* menjadi *Normalized Euclidean Distance* menggunakan **Persamaan 3.1** sebagai berikut:

$$N = \frac{n - X_{min}}{X_{max} - X_{min}}$$

$$N = \frac{158.7066 - 122.9824}{194.9772 - 122.9824}$$

$$N = \frac{35.7242}{71.9948}$$

$$N = 0.496205281$$

Untuk nilai *Normalized Euclidean Distance* 20 sampel lainnya dapat dilihat pada tabel dibawah ini:

Tabel 4.5 Nilai Normalized RGB

NO	DATA JERUK	NORMALISASI RED	NORMALISASI GREEN	NORMALISASI BLUE
1	Sampel 1	0.496205281	0.661126273	0.65338369
2	Sampel 2	0.603447749	0.769695204	1
3	Sampel 3	0.823646986	1	0.978948661
4	Sampel 4	0.575666576	0.638005949	0.517252108
5	Sampel 5	0.457108013	0.576060959	0.325192418
6	Sampel 6	0.862960936	0.764894824	0.736753869
7	Sampel 7	0.198140699	0.241208758	0.325472857
8	Sampel 8	0.155782084	0.197729363	0.325953902
9	Sampel 9	0.703917505	0.728415921	0.552583313
10	Sampel 10	0	0	0.145617375

11	Sampel 11	1	0.880711088	0.75176656
12	Sampel 12	0.410469923	0.500918374	0.49298084
13	Sampel 13	0.491904971	0.768723169	0.503283387
14	Sampel 14	0.596298622	0.79743346	0.715665684
15	Sampel 15	0.540061227	0.82444499	0.7741812
16	Sampel 16	0.140467089	0.637207163	0.858695652
17	Sampel 17	0.757017173	0.743572918	0.657414231
18	Sampel 18	0.702475734	0.74161045	0.601987636
19	Sampel 19	0.219375566	0.597838219	0.676690821
20	Sampel 20	0.085940929	0.076517846	0
21	Sampel 21	0.439416458	0.921757329	0.827982478

Penulis akan mencari nilai *Normalized Euclidean Distance* sampel jeruk dengan $k = 3$ menggunakan **Persamaan 3.2** sebagai berikut:

$$D(a,b,c) = \sqrt{\sum_{k=1}^d (a_k - b_k - c_k)^2}$$

$$D(a,b,c) = \sqrt{(0.496205281 - 0.439416458)^2 + (0.661126273 - 0.921757329)^2 + (0.65338369 - 0.827982478)^2}$$

$$D(a,b,c) = \sqrt{(0.056788823)^2 + (-0.260631056)^2 + (-0.174598788)^2}$$

$$D(a,b,c) = \sqrt{(0.00322497042) + (0.06792854735) + (0.03048473677)}$$

$$D(a,b,c) = \sqrt{0.10163825454}$$

$$D(a,b,c) = 0.318807551$$

Kemudian setelah nilai *Normalized Euclidean Distance* dihitung maka nilai tersebut akan diurutkan berdasarkan nilai yang paling terkecil. Untuk nilai *Normalized Euclidean Distance* 20 sampel lainnya dapat dilihat pada tabel di bawah ini:

Tabel 4.6 Normalized Euclidean Distance

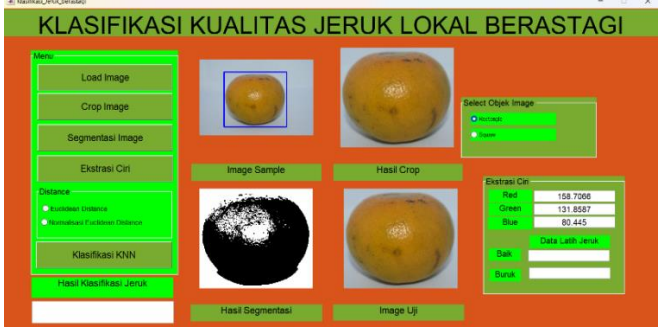

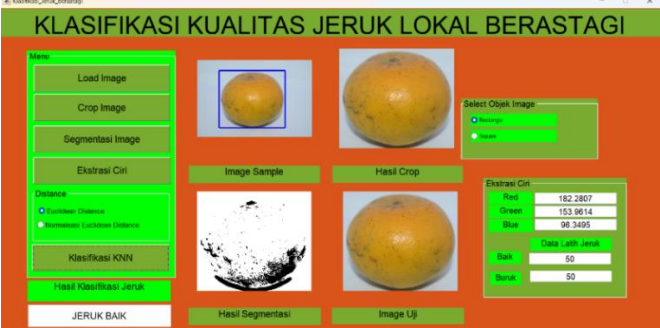

NORMALISASI EUCLIDEAN	URUTAN	KLASIFIKASI
0.318807551	4	BAIK
0.282168712	3	BURUK
0.420173539	10	BAIK
0.442303406	12	BAIK
0.610423472	16	BAIK
0.46078024	13	BURUK
0.879702375	17	BURUK
0.925579626	18	BURUK
0.428002808	11	BAIK
1.228147194	19	BURUK
0.567227968	15	BAIK
0.538673765	14	BAIK
0.362772653	5	BURUK
0.229528865	2	BAIK
0.149978793	1	BURUK
0.413863288	8	BURUK
0.402136078	7	BAIK
0.390802699	6	BAIK
0.419798462	9	BURUK
1.234880446	20	BURUK
0		

K = 3 BURUK

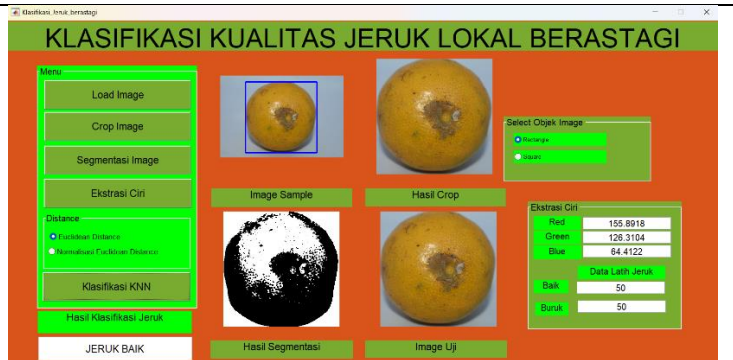
4.5 Hasil Pengujian

Pada penelitian ini, penulis melakukan pengujian terdapat 20 data gambar jeruk lokal Berastagi. Adapun hasil dari pengujian data tersebut dapat dilihat pada tabel di bawah ini:

Tabel 4.7 Hasil Pengujian

DATA JERUK	HASIL
Sampel 1	
Sampel 2	
Sampel 3	
Sampel 4	

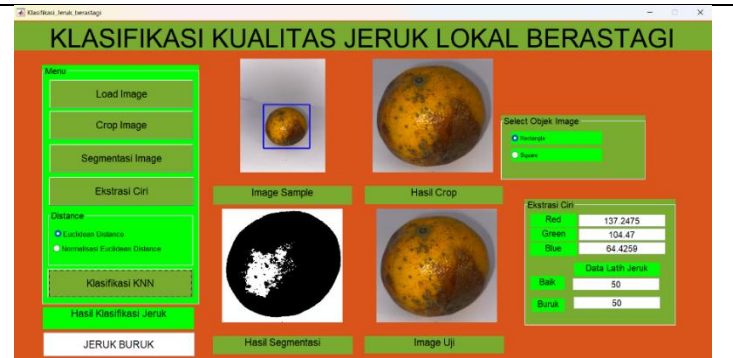
Sampel 5



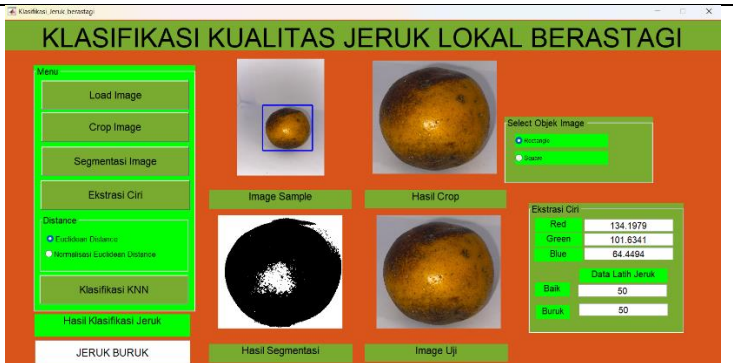
Sampel 6



Sampel 7



Sampel 8



Sampel 9



Sampel 10



Sampel 11



Sampel 12



Sampel 13



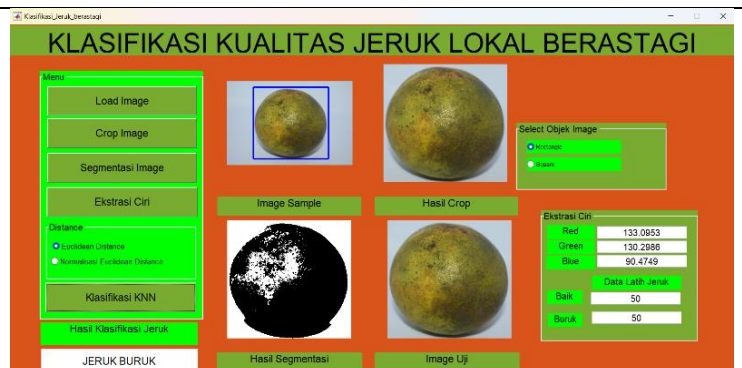
Sampel 14



Sampel 15



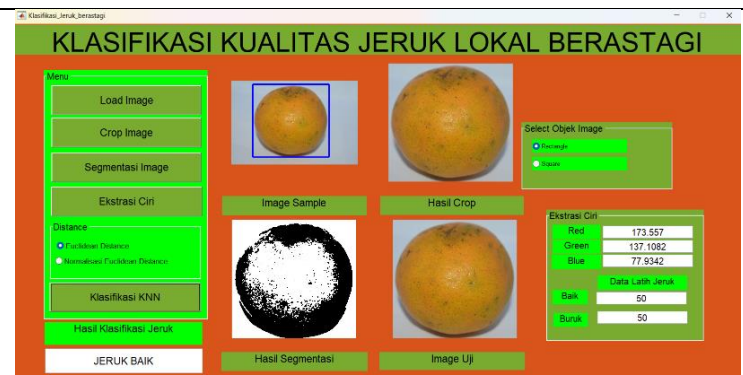
Sampel 16



Sampel 17



Sampel 18



Sampel 19



Sampel 20



Dari 20 sampel pengujian diatas, maka penulis mendapatkan hasil pengujian sebagai berikut:

- 1) Pada sampel 1, memiliki nilai red 158.7066, green 131.8587, blue 80.445 dengan hasil klasifikasi jeruk baik.
- 2) Pada sampel 2, memiliki nilai red 166.4275, green 138.94, blue 97.3779 dengan hasil klasifikasi jeruk buruk.
- 3) Pada sampel 3, memiliki nilai red 182.2807, green 153.9614, blue 96.3495 dengan hasil klasifikasi jeruk baik.
- 4) Pada sampel 4, memiliki nilai red 164.4274, green 130.3507, blue 73.7947 dengan hasil klasifikasi jeruk baik.
- 5) Pada sampel 5, memiliki nilai red 155.8918, green 120.3104, blue 64.4122 dengan hasil klasifikasi jeruk baik.
- 6) Pada sampel 6, memiliki nilai red 185.1111, green 138.6269, blue 84.5178 dengan hasil klasifikasi jeruk buruk.
- 7) Pada sampel 7, memiliki nilai red 137.2475, green 104.47, blue 64.4259 dengan hasil klasifikasi jeruk buruk.
- 8) Pada sampel 8, memiliki nilai red 134.1979, green 101.6341, blue 64.4494 dengan hasil klasifikasi jeruk buruk.
- 9) Pada sampel 9, memiliki nilai red 173.6608, green 135.2476, blue 75.5207 dengan hasil klasifikasi jeruk baik.
- 10) Pada sampel 10, memiliki nilai red 122.9824, green 88.7374, blue 55.6396 dengan hasil klasifikasi jeruk buruk.

- 11) Pada sampel 11, memiliki nilai red 194.9772, green 146.1809, blue 85.2512 dengan hasil klasifikasi jeruk baik.
- 12) Pada sampel 12, memiliki nilai red 152.5341, green 121.4093, blue 72.609 dengan hasil klasifikasi jeruk baik.
- 13) Pada sampel 13, memiliki nilai red 158.397, green 138.8766, blue 73.1123 dengan hasil klasifikasi jeruk buruk.
- 14) Pada sampel 14, memiliki nilai red 165.9128, green 140.7492, blue 83.4876 dengan hasil klasifikasi jeruk baik.
- 15) Pada sampel 15, memiliki nilai red 161.864, green 142.511, blue 86.3462 dengan hasil klasifikasi jeruk buruk.
- 16) Pada sampel 16, memiliki nilai red 133.0953, green 130.2986, blue 90.4749 dengan hasil klasifikasi jeruk buruk.
- 17) Pada sampel 17, memiliki nilai red 177.4837, green 137.2362, blue 80.6419 dengan hasil klasifikasi jeruk baik.
- 18) Pada sampel 18, memiliki nilai red 173.557, green 137.1082, blue 77.9342 dengan hasil klasifikasi jeruk baik.
- 19) Pada sampel 19, memiliki nilai red 138.7763, green 127.7308, blue 81.5836 dengan hasil klasifikasi jeruk buruk.
- 20) Pada sampel 20, memiliki nilai red 129.1697, green 93.7282, blue 48.5259 dengan hasil klasifikasi jeruk buruk.

Pada hasil pengujian yang peneliti lakukan diatas, peneliti mendapatkan tingkat akurasi 88% pada nilai ketetapan $K=3$.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan pada penelitian ini, dapat disimpulkan sebagai berikut:

1. Untuk mengklasifikasikan kualitas jeruk lokal Berastagi berdasarkan warna dari jeruk menggunakan *K-Nearest Neighbor* dimulai dari melakukan perhitungan *Euclidean Distance* dan *Normalized Euclidean Distance* untuk mengetahui nilai jarak antara 2 titik. Hasil klasifikasi jeruk lokal Berastagi berdasarkan dengan mencari label K tetangga terdekat.
2. Berdasarkan pengujian yang dilakukan bahwa tingkat akurasi *Euclidean Distance* menurun setelah dilakukan normalisasi, dikarenakan normalisasi hanya menjamin semua nilai akan memiliki skala yang sama yaitu 0-1 tetapi tidak menangani outlier dengan baik sehingga mempengaruhi kinerja *K-Nearest Neighbor* dalam menarik tetangga terdekat yang tidak relevan sehingga menyebabkan kesalahan dalam proses klasifikasi.
3. Metode klasifikasi *K-Nearest Neighbor* mampu mengklasifikasikan kualitas buah jeruk lokal Berastagi dengan tingkat akurasi sebesar 88% pada nilai ketetanggaan $K=3$.

5.2 Saran

Berdasarkan hasil yang diperoleh selama penelitian, penulis memberikan saran untuk peneliti berikutnya sebagai berikut:

1. Terdapat algoritma atau metode lain yang mungkin lebih akurat dan lebih singkat sehingga hasil yang diperoleh dapat lebih akurat.
2. Diharapkan untuk dikembangkan dalam berbasis mobile sehingga dapat digunakan langsung oleh para petani ketika mengambil sampel dilokasi dengan algoritma yang telah dikembangkan sebelumnya.

DAFTAR PUSTAKA

- Alfariza, E., & Chandra Artono, A. (2023). *TRAINER KIT SISTEM DISTRIBUSI DAN SORTIR BUAH JERUK BERDASARKAN BERAT DAN WARNA* (Doctoral dissertation, Politeknik Manufaktur Negeri Bangka Belitung).
- Al-Khowarizmi, R. S., Nasution, M. K., & Elveny, M. (2021). Sensitivity of MAPE using detection rate for big data forecasting crude palm oil on k-nearest neighbor. *Int. J. Electr. Comput. Eng*, *11*(3), 2696-2703.
- Al-Khowarizmi, A. K., & Suherman, S. (2021). Classification of skin cancer images by applying simple evolving connectionist system. *IAES International Journal of Artificial Intelligence*, *10*(2), 421.
- A'yuniyah, Q. A. Y., & Reza, M. (2023). Penerapan Algoritma K-Nearest Neighbor Untuk Klasifikasi Jurusan Siswa Di Sma Negeri 15 Pekanbaru: Application Of The K-Nearest Neighbor Algorithm For Student Department Classification At 15 Pekanbaru State High School. *Indonesian Journal of Informatic Research and Software Engineering (IJIRSE)*, *3*(1), 39-45.
- Elkashaf, N., & Hassan, Y. F. (2020). Mapping UML sequence diagram into the web ontology language OWL. *International Journal of Advanced Computer Science and Applications*, *11*(5).
- Fauzi, E., Sinatrya, M. V., Ramdhani, N. D., Ramadhan, R., & Safari, Z. M. R. (2022). Pengaruh kemajuan teknologi informasi terhadap perkembangan akuntansi. *Jurnal riset pendidikan ekonomi*, *7*(2), 189-197.

- Fauzi, J. R. (2020). Algoritma Dan Flowchart Dalam Menyelesaikan Suatu Masalah. *J. Tek. Inform*, 3(2), 12.
- Fajarini, I. P. N., Subroto, I. M. I., & Riansyah, A. (2022). Klasifikasi Kepakaran Reviewer Menggunakan Algoritma K-Nearest Neighbor (KNN). In *Prosiding Seminar Riset Mahasiswa (Vol. 1, No. 1, pp. 272-280)*.
- Fitriani, E. (2020). Perbandingan Algoritma C4. 5 Dan Naïve Bayes Untuk Menentukan Kelayakan Penerima Bantuan Program Keluarga Harapan. *SISTEMASI: Jurnal Sistem Informasi*, 9(1), 103-115.
- Hasibuan, M. P., Azmi, R., Arjuna, D. B., & Rahayu, S. U. (2023). Analisis pengukuran temperatur udara dengan metode observasi. *GABDIMAS: Jurnal Garuda Pengabdian Kepada Masyarakat*, 1(1), 8-15.
- Jayadi, B. V., Handhayani, T., & Lauro, M. D. (2023). Perbandingan Knn Dan Svm Untuk Klasifikasi Kualitas Udara Di Jakarta. *Jurnal Ilmu Komputer Dan Sistem Informasi*, 11(2).
- Kasanah, A. N., Muladi, M., & Pujianto, U. (2019). Penerapan Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Objektivitas Berita Online Menggunakan Algoritma KNN. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 3(2), 196-201.
- Maulana, S. (2021). KLASIFIKASI KUALITAS BUAH PIR BERDASARKAN WARNA DAN TEKSTUR DENGAN MENGGUNAKAN METODE K-

NEAREST NEIGHBOR (KNN) (Doctoral dissertation, Universitas Islam Lamongan).

Misar, A. R. P. PENERAPAN DAN PENGGUNAAN TEKNOLOGI MAJU BAGI INDUSTRI KREATIF DESAIN GRAFIS DALAM MENGHADAPI ASEAN ECONOMIC COMMUNITY 2015.

Nasution, D. A., Khotimah, H. H., & Chamidah, N. (2019). Perbandingan normalisasi data untuk klasifikasi wine menggunakan algoritma K-NN. *Comput. Eng. Sci. Syst. J*, 4(1), 78.

Novandya, A. (2017). Penerapan algoritma klasifikasi data mining C4. 5 pada dataset cuaca wilayah Bekasi. *Konferensi Nasional Ilmu Sosial dan Teknologi*, 1(1).

Pawening, R. E., Shudiq, W. J. F., & Wahyuni, W. (2020). Klasifikasi Kualitas Jeruk Lokal Berdasarkan Tekstur dan Bentuk Menggunakan Metode k-Nearest Neighbor (k-NN). *COREAI: Jurnal Kecerdasan Buatan, Komputasi dan Teknologi Informasi*, 1(1), 10-17.

Putra, R. T. (2023). *Optimasi Akurasi Sensor Arus Pzem-004t Melalui Penerapan Metode Jaringan Syaraf Tiruan* (Doctoral dissertation, Universitas Komputer Indonesia).

Rahayuningsih, P. A. (2019). Komparasi Algoritma Klasifikasi Data Mining untuk Memprediksi Tingkat Kematian Dini Kanker dengan Dataset Early Death Cancer. *JTIK (Jurnal Teknik Informatika Kaputama)*, 3(2), 1-10.

- Ridwan, M., Suyono, H., & Sarosa, M. (2013). Penerapan Data Mining Untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier. *jurnal EECCIS*, 7(1), 59-64.
- Sari, I. P., Azzahrah, A., Qathrunada, I. F., Lubis, N., & Anggraini, T. (2022). Perancangan sistem absensi pegawai kantor secara online pada website berbasis HTML dan CSS. *Blend sains jurnal teknik*, 1(1), 8-15.
- Sari, I. P., Batubara, I. H., Ramadhani, F., & Wardani, S. (2022). Perancangan Sistem Antrian pada Wahana Hiburan dengan Metode First In First Out (FIFO). *Sudo Jurnal Teknik Informatika*, 1(3), 116-123.
- Sari, I. P., Sulaiman, O. K., & Apdilah, D. (2024). Rancang Bangun Game Zombie Menggunakan Kodular Berbasis Android. *Jurnal Minfo Polgan*, 13(1), 293-302.
- Siwilopo, K. P., & Marcos, H. (2023). MEMBANDINGKAN KLASIFIKASI PADA BUAH JERUK MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK DAN K-NEAREST NEIGHBOR. *Komputa: Jurnal Ilmiah Komputer dan Informatika*, 12(1), 57-64.
- Sunandar, H. (2017). Perbaikan kualitas Citra Menggunakan Metode Gaussian Filter. *MEANS (Media Informasi Analisa dan Sistem)*, 19-22.
- Utomo, D. P., & Mesran, M. (2020). Analisis komparasi metode klasifikasi data mining dan reduksi atribut pada data set penyakit jantung. *Jurnal Media Informatika Budidarma*, 4(2), 437-444.

- Wahyudi, A., & Utami, R. (2022). Penggunaan Metode Euclidean Distance Pada Aplikasi Pencarian Lokasi Rumah Sakit di Kota Medan. *Informatics Engineering and Electronic Data (IEED)*, 1(1), 47-58.
- Yustanti, W. (2012). Algoritma K-Nearest Neighbour untuk Memprediksi Harga Jual Tanah. *Jurnal Matematika, Statistika dan Komputasi*, 9(1), 57-68.

DAFTAR LAMPIRAN

Lampiran I: Source Code Klasifikasi Jeruk Berastagi

```
function varargout = Klasifikasi_Jeruk_berastagi(varargin)

gui_Singleton = 1;

gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @Klasifikasi_Jeruk_berastagi_OpeningFcn, ...
                  'gui_OutputFcn', @Klasifikasi_Jeruk_berastagi_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function Klasifikasi_Jeruk_berastagi_OpeningFcn(hObject, eventdata, handles,
varargin)

function varargout = Klasifikasi_Jeruk_berastagi_OutputFcn(hObject, eventdata,
handles)
```

```
function edit1_Callback(hObject, eventdata, handles)
function edit1_CreateFcn(hObject, eventdata, handles)
function edit2_Callback(hObject, eventdata, handles)
function edit2_CreateFcn(hObject, eventdata, handles)
function edit3_Callback(hObject, eventdata, handles)
function edit3_CreateFcn(hObject, eventdata, handles)
function edit4_Callback(hObject, eventdata, handles)
function edit4_CreateFcn(hObject, eventdata, handles)
function edit5_Callback(hObject, eventdata, handles)
function edit5_CreateFcn(hObject, eventdata, handles)
function edit6_Callback(hObject, eventdata, handles)
function edit6_CreateFcn(hObject, eventdata, handles)
function pushbutton2_Callback(hObject, eventdata, handles)

axes(handles.axes1)

cla reset

set(gca,'XTick',[])
set(gca,'YTick',[])

axes(handles.axes2)

cla reset

set(gca,'XTick',[])
set(gca,'YTick',[])

axes(handles.axes3)
```

```
cla reset

set(gca,'XTick',[])

set(gca,'YTick',[])

axes(handles.axes4)

cla reset

set(gca,'XTick',[])

set(gca,'YTick',[])

set(handles.radiobutton4,'Value',0)

set(handles.radiobutton5,'Value',0)

set(handles.radiobutton1,'Value',0)

set(handles.radiobutton2,'Value',0)

set(handles.edit1,'String',[])

set(handles.edit2,'String',[])

set(handles.edit3,'String',[])

set(handles.edit4,'String',[])

set(handles.edit5,'String',[])

    set(handles.edit6,'String',[])

[filename,pathname] = uigetfile({'*.*'});

if isequal(filename,0)

    return

else

    Info = imfinfo(fullfile(pathname,filename));
```

```

BitDepth = Info.BitDepth;

if BitDepth == 1

    msgbox('Please insert RGB image or Grayscale image')

    return

end

Img = imread(fullfile(pathname,filename));

axes(handles.axes1)

cla reset

set(gca,'XTick',[])

set(gca,'YTick',[])

imshow(Img)

end

set(handles.radiobutton1,'Enable','on')

set(handles.radiobutton2,'Enable','on')

handles.Img = Img;

handles.BitDepth = BitDepth;

guidata(hObject, handles)

function pushbutton3_Callback(hObject, eventdata, handles)

Img = handles.Img;

[m,n,~] = size(Img);

val1 = get(handles.radiobutton1,'Value');

val2 = get(handles.radiobutton2,'Value');

```

```

axes(handles.axes2)

cla reset

set(gca,'XTick',[])

set(gca,'YTick',[])

axes(handles.axes1)

cla reset

set(gca,'XTick',[])

set(gca,'YTick',[])

imshow(Img);

if val1 == 1

    h = imrect(gca,[n/2 m/2 0.1*n 0.1*m]);

    wait(h);

    mask = createMask(h);

elseif val2 == 1

    h = imrect(gca,[n/2 m/2 0.1*m 0.1*m]);

    setFixedAspectRatioMode(h,'true')

    wait(h);

    mask = createMask(h);

end

[B,~] = bwboundaries(mask,'noholes');

for k = 1:length(B)

    boundary = B{k};

```

```

axes(handles.axes1)

cla reset

set(gca,'XTick',[])

set(gca,'YTick',[])

imshow(Img);

hold on

plot(boundary(:,2), boundary(:,1), 'b', 'LineWidth', 2)

hold off

end

if handles.BitDepth == 10

    Img(~mask) = 0;

    [row,col] = find(mask==1);

    h_Gray = imcrop(Img,[min(col) min(row) max(col)-min(col) max(row)-
min(row)]);

    axes(handles.axes2)

    imshow(h_Gray);

    imwrite(h_Gray,'data.jpg')

else

    R = Img(:,:,1);

    G = Img(:,:,2);

    B = Img(:,:,3);

    R(~mask) = 0;

```



```

G(~mask) = 0;

B(~mask) = 0;

RGB = cat(3,R,G,B);

[row,col] = find(mask==1);

h_RGB = imcrop(RGB,[min(col) min(row) max(col)-min(col) max(row)-
min(row)]);

axes(handles.axes2)

imshow(h_RGB);

imwrite(h_RGB,'data.jpg')

end

```

Share this:

```

axes(handles.axes3)

cla reset

set(gca,'XTick',[])

set(gca,'YTick',[])

function pushbutton4_Callback(hObject, eventdata, handles)

I= imread('data.jpg');

%mengkonversi citra rgb menjadi citra grayscale

Img_gray = rgb2gray(I) ;

mf = medfilt2(Img_gray, [5 5]);

bk = im2bw(mf);

axes(handles.axes3)

```

```
imshow(bk)

bw = imbinarize(Img_gray,9);

bw = imcomplement(bw);

bw = imfill(bw,'holes');

bw = bwareaopen(bw,100);

R = I (:,:,1);

G = I (:,:,2);

B = I (:,:,3);

R(~bw) = 0;

G(~bw) = 0;

B(~bw) = 0;

RGB = cat(3,R,G,B);

% menampilkan citra RGB hasil segmentasi pada axes4

axes(handles.axes4)

imshow(RGB)

%menyimpan variable R,G,B dan bw pada lokasi handles agar dapat

%dipanggil

%oelh pushbutton lainnya

handles.R = R;

handles.G = G;

handles.B = B;

handles.bw = bw;
```

```

guidata(hObject, handles)

function pushbutton5_Callback(hObject, eventdata, handles)

val3 = get(handles.radiobutton5,'Value');

val4 = get(handles.radiobutton4,'Value');

if val3 == 1

    [num_train,txt,row] = xlsread('data_trainjeruk.xlsx',1);

    [num_kelas,kelas,row_kelas] = xlsread('data_trainjeruk.xlsx',2);

    train = fitcknn(num_train,kelas);

    train.Distance = 'euclidean';

    R = handles.R;

    G = handles.G;

    B = handles.B;

    bw = handles.bw;

    Red = sum(sum(R))/sum(sum(bw));

    Green = sum(sum(G))/sum(sum(bw));

    Blue = sum(sum(B))/sum(sum(bw));

    %menyusun variable ciri_uji

    ciri_uji = [Red,Green,Blue];

    %Prediksi data train dengan data image

    hasil = predict(train,ciri_uji);

    disp(hasil());

    set(handles.edit5,'String',50)

```

```

set(handles.edit6,'String',50)

set(handles.edit1,'String',hasil())

elseif val4 == 1

    [num_train,txt,row] = xlsread('data_trainjeruknorma.xlsx',1);

[num_kelas,kelas,row_kelas] = xlsread('data_trainjeruknorma.xlsx',2);

train = fitcknn(num_train,kelas);

train.Distance = 'euclidean',3;

R = handles.R;

G = handles.G;

B = handles.B;

bw = handles.bw;

Red = sum(sum(R))/sum(sum(bw));

Green = sum(sum(G))/sum(sum(bw));

Blue = sum(sum(B))/sum(sum(bw));

nRed = (Red-124.0334)/(9195.2832-124.0334);

nGreen = (Green-90.5066)/(154.4344-90.5066);

nBlue = (Blue-48.8399)/(98.9037-48.8399);

ciri_uji = [nRed,nGreen,nBlue];

hasil = predict(train,ciri_uji);

disp(hasil());

set(handles.edit5,'String',50)

set(handles.edit6,'String',50)

```

```

set(handles.edit1,'String',hasil())

end

function checkbox1_Callback(hObject, eventdata, handles)

function checkbox2_Callback(hObject, eventdata, handles)

function radiobutton1_Callback(hObject, eventdata, handles)

set(handles.radiobutton1,'Value',1)

set(handles.radiobutton2,'Value',0)

function radiobutton2_Callback(hObject, eventdata, handles)

set(handles.radiobutton1,'Value',0)

set(handles.radiobutton2,'Value',1)

function pushbutton6_Callback(hObject, eventdata, handles)

R = handles.R;

G = handles.G;

B = handles.B;

bw = handles.bw;

Red = sum(sum(R))/sum(sum(bw));

Green = sum(sum(G))/sum(sum(bw));

Blue = sum(sum(B))/sum(sum(bw));

%menyusun variable ciri_uji

ciri_uji = [Red,Green,Blue];

% menampilkan ciri pada edit text

set(handles.edit2,'string',num2str(Red));

```

```

set(handles.edit3,'string',num2str(Green));

set(handles.edit4,'string',num2str(Blue));

%menyimpan variable ciri_uji pada lokasi handles agar dapat dipanggil oleh

%pushbutton lain

handles.ciri_uji = ciri_uji;

guidata(hObject,handles)

function radiobutton3_Callback(hObject, eventdata, handles)

function radiobutton4_Callback(hObject, eventdata, handles)

set(handles.radiobutton5,'Value',0)

set(handles.radiobutton4,'Value',1)

function radiobutton5_Callback(hObject, eventdata, handles)

set(handles.radiobutton5,'Value',1)

set(handles.radiobutton4,'Value',0)

```

Lampiran II: Source Code KNN Jeruk

```

[num_train,txt,row] = xlsread('data_trainn.xlsx',1);

[num_kelas,kelas,row_kelas] = xlsread('data_trainn.xlsx',2);

train = fitcknn(num_train,kelas);

img = imread('BA7.jpg');

figure,imshow(img);

grayA1 = rgb2gray(img);

mfA1 = medfilt2(grayA1, [5 5]);

bwA1 = im2bw(mfA1);

```

```

figure,imshow(bw)

SSEdis1= strel('disk', 10, 0);

SEdis2= strel('disk', 5, 0 );

SEdis3= strel('disk', 1, 0);

eroA1 = imerode(bwA1, SEdis2);

reA1 = imcomplement(eroA1);

cc = bwconncomp(reA1,4);

adaun(cc.PixelIdxList{1}) = true;

fitur = regionprops(cc,'Area','MajorAxisLength','MinorAxisLength',
'Eccentricity','Perimeter');

data = [fitur(3).Area fitur(3).MajorAxisLength fitur(3).MinorAxisLength
fitur(3).Eccentricity fitur(3).Perimeter];

disp (data);

%Prediksi data train dengan data image

hasil = predict(train,data);

disp(hasil());

```

Lampiran III: Source Code Latih Jeruk

```

% membaca file citra

nama_folder = 'jeruk baik'

nama_file = dir(fullfile(nama_folder, '*.jpg'));

jumlah_file = numel(nama_file);

%menginisialisasi variabel ciri baik dan target _baikpoe

```

```

ciri_baik = zeros (jumlah_file, 50);

target_baik = cell(jumlah_file, 1) ;

%melakukan pengolahan citra terhadap seluruh file

for n = 1:jumlah_file

    %membaca file citra rgb

    Img = im2double(imread(fullfile(nama_folder, nama_file(n).name))) ;

    nama_file(n).name

    %mengkonversi citra rgb menjadi citra grayscale

    Img_gray = rgb2gray(Img) ;

    %figure, imshow(Img)

    %figure, imshow(Img_gray)

    %konversi citra greyscale menjadi biner

    bw = imbinarize(Img_gray,.9);

    %figure, imshow(bw)

    %melakukan operasi komplemen

    bw = imcomplement(bw);

    %figure, imshow(bw)

    %melakukan operasi morfologi

    %(1) filling holes

    bw = imfill(bw,'holes');

    %figure, imshow(bw)

    %(2) area opening

```



```

bw = bwareaopen(bw,100);

%figure, imshow(bw)

%ekstrasi warna rgb

R = Img (:,:,1);

G = Img (:,:,2);

B = Img (:,:,3);

R(~bw) = 0;

G(~bw) = 0;

B(~bw) = 0;

RGB = cat(3,R,G,B);

%figure, imshow(RGB)

Red = sum(sum(R))/sum(sum(bw));

Green = sum(sum(G))/sum(sum(bw));

Blue = sum(sum(B))/sum(sum(bw));

%mengisi variable ciri_baik dengan hasil ekstraksi

ciri_baik(n,1) = Red;

ciri_baik(n,2) = Green;

ciri_baik(n,3) = Blue;

%mengisi variable ke targer_baik

target_baik{n} = 'jeruk baik';

end

% jeruk buruk

```

```

% membaca file citra

nama_folder = 'jeruk buruk'

nama_file = dir(fullfile(nama_folder, '*.jpg'));

jumlah_file = numel(nama_file);

%menginisialisasi variabel ciri buruk dan target_buruk

ciri_buruk = zeros (jumlah_file, 50);

target_buruk = cell(jumlah_file, 1) ;

%melakukan pengolahan citra terhadap seluruh file

for n = 1:jumlah_file

    %membaca file citra rgb

    Img = im2double(imread(fullfile(nama_folder, nama_file(n).name))) ;

    nama_file(n).name

    %mengkonversi citra rgb menjadi citra grayscale

    Img_gray = rgb2gray(Img) ;

    %figure, imshow(Img)

    %figure, imshow(Img_gray)

    %konversi citra greyscale menjadi biner

    bw = imbinarize(Img_gray,.9);

    %figure, imshow(bw)

    %melakukan operasi komplemen

    bw = imcomplement(bw);

    %figure, imshow(bw)

```

```

%melakukan operasi morfologi

%(1) filling holes

bw = imfill(bw,'holes');

%figure, imshow(bw)

%(2) area opening

bw = bwareaopen(bw,100);

%figure, imshow(bw)

%ekstraksi warna rgb

R = Img (:,:,1);

G = Img (:,:,2);

B = Img (:,:,3);

R(~bw) = 0;

g(~bw) = 0;

B(~bw) = 0;

RGB = cat(3,R,G,B);

%figure, imshow(RGB)

Red = sum(sum(R))/sum(sum(bw));

Green = sum(sum(G))/sum(sum(bw));

Blue = sum(sum(B))/sum(sum(bw));

%mengisi variable ciri_buruk dengan hasil ekstraksi

ciri_buruk(n,1) = Red;

ciri_buruk(n,2) = Green;

```

```

    ciri_buruk(n,3) = Blue;

    %mengisi variable ke target_buruk

    target_buruk{n} = 'jeruk buruk';

end

%menyusun variable ciri_latih dan target_latih

ciri_latih = [ciri_baik;ciri_buruk];

target_latih = [target_baik;target_buruk];

%melakukan pelatihan dengan K-NN

Mdl = fitcknn(ciri_latih,target_latih,'NumNeighbors',3);

%membaca kelas keluaran hasil pelatihan

hasil_latih = predict(Mdl,ciri_latih);

%menghitung akurasi pelatihan

jumlah_benar = 0;

jumlah_data = size(ciri_latih,1);

for k = 1:jumlah_data

    if isequal(hasil_latih{k}, target_latih {k})

        jumlah_benar = jumlah_benar+1;

    end

end

akurasi_pelatihan = jumlah_benar/jumlah_data*100

%menyimpan variable Mdl hasil Pelatihan

save Mdl Mdl

```

