



PENERAPAN ALGORITMA K-NEAREST NEIGHBOR UNTUK KLASIFIKASI KEMATANGAN BUAH JERUK NIPIS

Yusril Fahmi¹, Qomaruzzaman², Soffiana Agustin³

^{1,2,3}Teknik Informatika, Teknik, Universitas Muhammadiyah Gresik

¹ysrfahmi@gmail.com, ²qomaruzzaman.1708@gmail.com, ³soffiana@umg.ac.id

Abstract

Classification of fruit ripeness has become an important topic in agriculture, as this process often requires considerable time and effort. This study aims to develop an automatic classification system that can identify the ripeness level of lime (unripe, half-ripe, ripe) based on RGB and HSV color features using the K-Nearest Neighbor (KNN) algorithm. A total of 83 datasets were collected using a Poco M3 Pro 5G camera, followed by preprocessing, feature extraction, and classification with the KNN algorithm. Using 16 test data in classification, the highest accuracy achieved was 75% with k=5. The implementation of this method demonstrates that KNN is quite effective in classifying color features.

Keywords: lime, k-Nearest Neighbor, digital image.

Abstrak

Klasifikasi pada kematangan buah menjadi kajian penting dalam bidang pertanian, lantaran kegiatan tersebut seringkali memakan banyak waktu dan tenaga. Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi otomatis yang dapat mengidentifikasi kematangan buah jeruk nipis (mentah, setengah matang, matang) berdasarkan fitur warna RGB dan HSV dengan algoritma K-Nearest Neighbor (KNN). Terdapat 83 dataset yang diambil dengan Poco M3 Pro 5G, lalu diproses dalam preprocessing, ekstraksi fitur dan di klasifikasi dengan algoritma KNN. Menggunakan 16 data uji dalam klasifikasi, mendapati hasil akurasi terbesar di 75% menggunakan k=5. Implementasi metode ini menunjukkan menunjukkan bahwa KNN cukup efektif dalam melakukan klasifikasi ciri warna.

Kata kunci: jeruk nipis, k-nearest neighbor, citra digital.

1. Pendahuluan

Buah-buahan menjadi salah satu produk pangan alami yang paling banyak diminati di Indonesia. Dengan kandungan vitamin yang baik untuk tubuh membuat buah-buahan memiliki tingkat permintaan yang tinggi dan diminati oleh semua kalangan umur. Permintaan yang tinggi terhadap buah-buahan dapat dilihat di pasar tradisional yang sebagian transaksinya diisi oleh sektor tersebut, contohnya permintaan buah seperti jeruk, apel, pisang dan lain-lainnya. Tingginya permintaan masyarakat Indonesia pada buah-buahan membuat para petani harus mampu mengklasifikasikan tingkat kesegaran dan kematangan buah. Petani dapat melakukan secara manual dalam melakukan klasifikasi buah-buahan seperti jeruk, pisang dan apel.

Namun, mata manusia terkadang tidak akurat dengan adanya gangguan dari berbagai faktor, seperti bertambahnya minus, silinder maupun rabun. Faktor tenaga dan waktu juga perlu dipertimbangkan dalam

melakukan klasifikasi buah-buahan[1]. Dalam melakukan klasifikasi buah-buahan dengan tujuan utama pada buah jeruk nipis, memungkinkan untuk dikenali dengan baik melalui warna. Kematangan buah jeruk nipis pada umumnya dapat dibedakan melalui warna buah. Proses pengklasifikasian kematangan buah yang berfokus pada warna dapat dilakukan dengan ekstraksi fitur RGB dan HSV dengan menggunakan metode K-Nearest Neighbor[2].

Penelitian ini bertujuan untuk mengetahui apakah metode K-Nearest Neighbor dapat digunakan untuk melakukan klasifikasi terhadap kematangan buah jeruk nipis, dengan mengenali data warna maka nantinya diharapkan dapat mengenali kematangan buah yang akan diuji.

Ekstraksi warna dapat diandalkan dalam kasus menilai kematangan buah lantaran dapat menjelaskan warna berdasarkan terminologi masing-masing kanal. RGB dapat menjelaskan warna berdasarkan terminologi

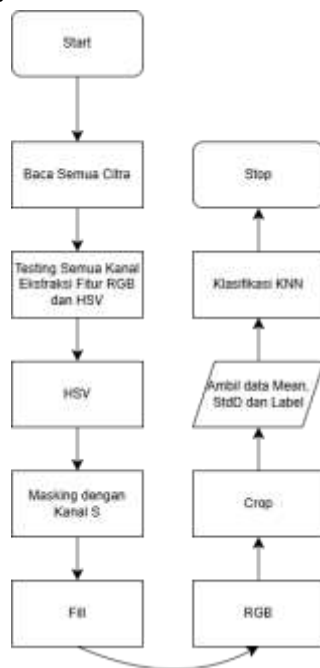
Red, Green, Blue. HSV dapat menjelaskan warna berdasarkan terminologi Hue, Saturation, Value. Dalam kasus menilai kematangan buah jeruk nipis, tingkat kematangan buah memang dapat dilihat lantaran mengalami perubahan warna, dengan begitu maka cocok ketika menggunakan ekstraksi warna. Setelah nantinya akan mendapatkan data warna dari setiap terminologi, maka dapat diklasifikasikan dan dapat diuji dengan metode K-Nearest Neighbors.

Prinsip K-Nearest Neighbors mengklasifikasikan data uji berdasarkan kemiripan atau jarak terdekat dari data training. Metode K-Nearest Neighbors dipilih lantaran tahan terhadap data training yang memiliki noise dan efektif[2]. Melalui studi literatur juga didapati dalam kasus mengklasifikasikan buah belimbing yang proses kematangannya juga sama mengalami perubahan warna buah, didapati bahwa metode KNN dapat mengklasifikasikan dengan baik[3].

Dengan adanya penelitian ini, diharapkan dapat memberikan informasi kepada para petani buah jeruk nipis agar dapat diterapkan guna mengoptimalkan pengklasifikasikan kematangan buah. Diharapkan juga dapat memberikan data aktual mengenai metode KNN dengan ekstraksi warna RGB dan HSV dalam melakukan klasifikasi kematangan buah khususnya jeruk nipis.

2. Metode Penelitian

Pengerjaan dalam penelitian ini melewati berbagai proses, mulai dari pengumpulan dataset yang dilakukan secara mandiri, lalu diolah melalui preprocessing, ekstraksi fitur dan nantinya masuk dalam klasifikasi yang akan dilakukan menggunakan metode K-Nearest Neighbors, berikut proses penelitiannya:



Gambar 1. Alur sistem

Penelitian ini keseluruhan menggunakan dataset pribadi, dengan berbagai warna latar belakang dan dibedakan dalam setiap folder sesuai dengan kematangannya, selain data yang akan diuji. Hasil dari klasifikasi ini didasarkan pada ciri warna, dengan menggunakan metode KNN untuk melakukan klasifikasi jeruk nipis menjadi tiga jenis diantaranya mentah, setengah matang, dan matang.

Pengambilan dataset untuk penelitian ini menggunakan peralatan kamera smartphone Poco M3 Pro 5G dan tripod. Kamera smartphone Poco M3 Pro 5G dengan resolusi kamera 48 MP digunakan untuk mengambil gambar dan menempatkan objek di tengah frame kamera dengan berbagai background berbeda. Dengan bantuan tripod, maka meminimalisir pergeseran posisi foto dalam tempat yang sama, dan dengan jarak kamera ke objek yang konsisten di kurang lebih 35 centimeter.

Data dikumpulkan dalam folder terpisah menjadi tiga jenis: mentah, setengah matang, matang. Dengan total 83 citra terdiri dari 19 citra jeruk nipis mentah, 21 citra jeruk nipis setengah matang, 27 citra jeruk nipis matang dan 16 citra jeruk nipis uji. Contoh data jenis kematangan buah jeruk nipis dapat dilihat pada gambar 2.



Gambar 2. Contoh data jenis kematangan buah

Memasuki preprocessing, tahap preprocessing citra adalah tahap penting untuk memastikan kualitas citra yang terbaik untuk selanjutnya diproses dalam ekstraksi fitur, dalam tahap ini citra awal akan dimaksimalkan kualitasnya[4]. Preprocessing adalah suatu proses mempengaruhi langsung dalam ekstraksi fitur dan klasifikasi, dalam proses ini objek utama dapat dipisahkan dari latar belakangnya, dapat mengubah tampilan noise menjadi bersih, dan lain-lain sesuai dengan kebutuhan dari citra awal.

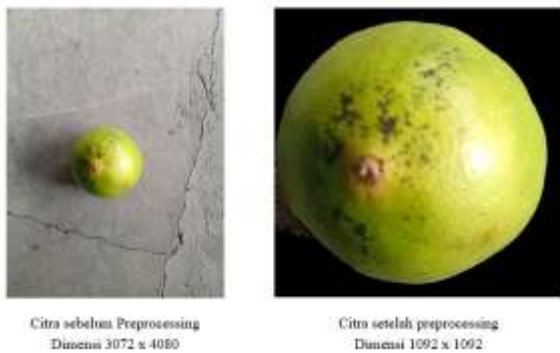
Pada penelitian ini, dibutuhkan preprocessing untuk melakukan cropping pada citra dan segmentasi atau memisahkan objek utama dengan latar belakangnya. Dalam prosesnya, akan bertemu dengan ruang warna RGB dan HSV yang dapat memudahkan pengaturan warna di sistem koordinat tiga dimensi. Dengan mempartisi citra menjadi wilayah-wilayah yang bermakna, segmentasi memungkinkan penggunaan region yang diinginkan, atau biasa disebut dengan Region of Interest (ROI). Pengelompokan piksel dalam citra menjadi unit homogen dengan kesamaan warna, tekstur, maupun intensitas disebut proses segmentasi[5].

Menggunakan ruang warna RGB dan HSV, setiap kanal dari ekstraksi warna dilakukan uji coba thresholding terhadap dataset guna mengetahui kanal mana yang paling cocok untuk digunakan dalam proses pemisahan objek utama dengan latar belakang. Setelah dilakukan uji coba didapati bahwasannya kanal S pada HSV dapat mendeteksi objek utama dengan sangat baik dibanding lainnya, maka akan digunakan dalam proses thresholding.



Gambar 3. Hasil Masking

Dengan thresholding yang berjalan di HSV, maka citra akan diubah dahulu ke HSV dan membuat citra menjadi abu-abu, lalu dilakukan thresholding dan diproses dengan fill agar objek utama yang sekelilingnya terkena dampak threshold maka akan diisi kembali, setelah proses tersebut selesai maka akan citra akan dikembalikan lagi ke RGB dan dilakukan cropping agar ketika melakukan ekstraksi fitur, nilai-nilai yang didapat dari citra akan berfokus kepada objek utama.



Gambar 4. Perubahan hasil Proprocessing

Masuk di tahap ekstraksi fitur, ekstraksi fitur digunakan untuk mendapatkan nilai fitur dari suatu objek berdasarkan hubungan nilai intensitas piksel suatu citra [6][7]. Pada penelitian ini, ekstraksi fitur mendapatkan nilai fitur dari citra melalui mean RGB dan HSV, lantaran kematangan pada buah jeruk nipis dapat terlihat berbeda mulai dari mentah hingga matang. Tujuan dari ekstraksi fitur dalam penelitian ini adalah untuk klasifikasi berbagai variasi citra buah

jeruk nipis, nantinya setiap kanal RGB dan HSV akan dicatat nilai mean dan standard deviasi untuk menjadi data dalam proses klasifikasi. Adapun rumus mean dan standard deviasi yang disajikan sebagai berikut:

Mean:

$$\mu = \frac{1}{N} \sum_{i=1}^N A_i \quad (1)$$

N = jumlah data

A_i= nilai fitur

Standard Deviasi:

$$S = \sqrt{\frac{1}{N-1} \sum_{i=1}^N |A_i - \mu|^2} \quad (2)$$

N = jumlah data

A_i= nilai fitur

μ = mean

Channel	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0000	0.4670	0.1833	0.1173	0.1190	0.0246	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.1173	0.4340	0.1480	0.1133	0.1433	0.0400	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.2680	0.2680	0.2370	0.0833	0.1100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
4	0.0000	0.4333	0.1667	0.1177	0.1122	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5	0.1480	0.4011	0.2000	0.2383	0.2007	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.0400	0.0400	0.2780	0.2000	0.0000	0.1100	0.1119	0.0000	0.0000	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.0000	0.0000	0.1133	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
12	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
13	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
14	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
16	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
17	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
18	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
19	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
20	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
21	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
22	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
23	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
24	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
25	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
26	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
27	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
28	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
29	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
30	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
31	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
32	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
33	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
34	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
35	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
36	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
37	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
38	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
39	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
40	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
41	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
42	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
43	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
44	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
45	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
46	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
47	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
48	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
49	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Gambar 5. Hasil Ekstraksi fitur mean & standard deviasi RGB dan HSV

Langkah terakhir yaitu klasifikasi, klasifikasi dipergunakan untuk mengambil keputusan berdasarkan data yang ada dengan menggunakan algoritma tertentu[5]. Dalam penelitian ini, diputuskan untuk menggunakan metode K-Nearest Neighbor (KNN) untuk melakukan proses klasifikasi lantaran sederhana dalam konsep dan memiliki hasil yang cukup baik, KKN bekerja dengan cara membandingkan data uji

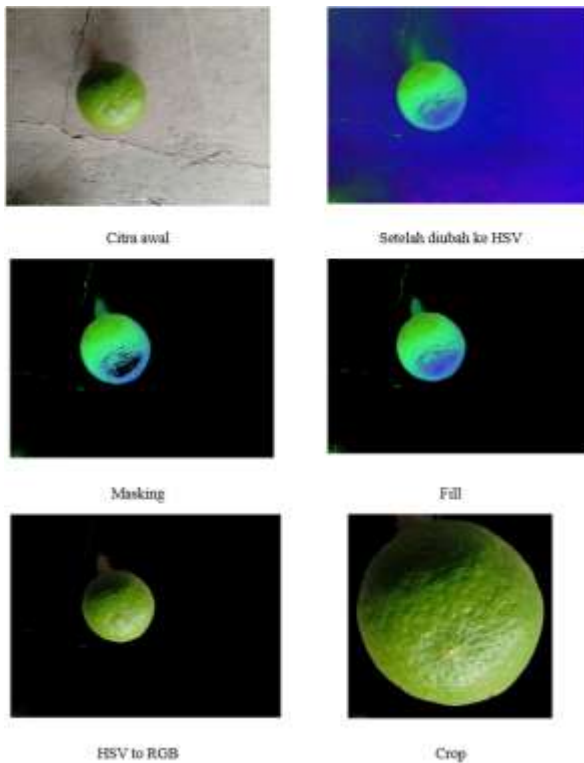
dan data training, dengan mencari jarak terdekat data training yang mendekati data uji[8][9][10].

Dalam langkahnya, diperlukan persiapan data dengan memastikan semua citra telah melalui proses preprocessing maupun ekstraksi fitur, dengan begitu maka didapatkan data latih dan data uji. Lalu penentuan K (jumlah tetangga) sebagai data dari metode K-Nearest Neighbor (KNN) saat mengelola dataset.

3. Hasil dan Pembahasan

Proses penelitian kematangan buah jeruk nipis memanfaatkan metode K-Nearest Neighbor (KNN) untuk membedakan tiga label kematangan: mentah, setengah matang, matang. Dengan total 83 citra terbagi dari 19 citra jeruk nipis mentah, 21 citra setengah matang, 27 citra matang dan 16 citra untuk menjadi data uji. Proses yang dilakukan sebagai berikut:

Dengan citra yang ada, dibutuhkan lima tahapan untuk memisahkan dan mengoptimalkan citra data training agar dapat di ekstraksi fitur dengan baik. Prosesnya diawali dengan melakukan perubahan citra RGB menjadi HSV agar citra menjadi abu-abu, lalu dilanjutkan dengan melakukan thresholding untuk memisahkan objek utama dengan latar belakang, dilanjutkan dengan melakukan fill agar objek utama yang terkena dampak proses thresholding dapat disempurnakan kembali, setelah citra dirasa sudah maksimal maka dikembalikan ke RGB untuk mendapatkan warna asli dari citra, dan lakukan crop untuk memfokuskan citra ke objek utama.



Gambar 6. Output tahap proses preprocessing

Dalam proses ekstraksi fitur, citra yang sudah di maksimalkan dalam preprocessing lalu diekstraksi sebagai indikator karakteristik dalam bentuk nilai. Pada penelitian ini, indikator yang digunakan adalah mean dan standard deviasi dari RGB dan HSV, lalu data akan digunakan dalam proses klasifikasi.

Dalam proses klasifikasi, algoritma K-Nearest Neighbor (KNN) bertugas untuk memprediksi jenis kematangan buah dari citra uji, untuk melakukan klasifikasi diperlukan untuk menentukan nilai k (tetangga terdekat). Nantinya, algoritma K-Nearest Neighbor (KNN) akan menghitung jarak data uji ke data latih dan ditentukan tetangga terdekatnya berdasarkan jarak yang telah dihitung, setelah didapatkan posisi data uji dari serangkaian data latih maka dilakukan majority voting untuk menentukan kelas dari citra uji berdasarkan mayoritas dari tetangga terdekat[11].

Dengan 16 citra data uji, tabel 2 menunjukkan hasil dari proses klasifikasi yang dimana dalam prosesnya menggunakan k=5, terdapat 4 citra yang salah diprediksi, dan 12 citra lainnya benar dengan 3 kelas yaitu: mentah, setengah matang dan matang.

Tabel 1. Hasil KNN

Nama Data Uji	Jenis Kematangan	Hasil Prediksi	Benar / Salah
Tes 1	Setengah Matang	Setengah Matang	Benar
Tes 2	Mentah	Mentah	Benar
Tes 3	Matang	Matang	Benar
Tes 4	Mentah	Mentah	Benar
Tes 5	Matang	Matang	Benar
Tes 6	Setengah Matang	Setengah Matang	Benar
Tes 7	Mentah	Mentah	Benar
Tes 8	Setengah Matang	Mentah	Salah
Tes 9	Setengah Matang	Mentah	Salah
Tes 10	Setengah Matang	Mentah	Salah
Tes 11	Setengah Matang	Mentah	Salah
Tes 12	Setengah Matang	Setengah Matang	Benar
Tes 13	Setengah Matang	Setengah Matang	Benar
Tes 14	Setengah Matang	Setengah Matang	Benar
Tes 15	Setengah Matang	Setengah Matang	Benar
Tes 16	Setengah Matang	Setengah Matang	Benar

Penelitian ini melakukan percobaan pada beberapa kondisi k, mulai dari k=1, k=3, k=5, k=7 dan k=9. Didapati pada k=1, k=3, dan k=5 hasil akurasi kebenaran yang sama yaitu 75%, sedangkan k=7 dan k=9 mendapatkan akurasi 68.75%.

Tabel 2. Hasil Uji Coba K

K	Akurasi
1	75.00%
3	75.00%
5	75.00%
7	68.75%
9	68.75%

4. Kesimpulan

Menurut penelitian yang telah dilakukan, fitur warna yang digunakan dalam mengidentifikasi kematangan buah jeruk nipis dengan metode K-Nearest Neighbor (KNN) dapat berjalan dengan baik. Dalam tabel 3 menunjukkan dengan metode K-Nearest Neighbor (KNN) didapati hasil akurasi tertinggi sebesar 75% dengan nilai k=1, k=3, dan k=5. Hasil penelitian ini membuktikan bahwa metode K-Nearest Neighbor (KNN) dapat diandalkan dan cukup baik dalam mengklasifikasikan kematangan buah jeruk nipis berdasarkan ciri warna.

Daftar Rujukan

- [1] S. Napitu, R. Paramita Panjaitan, P. A. Nulhakim, and M. Khalik Lubis, "Klasifikasi Buah Jeruk Segar dan Busuk Berdasarkan RGB dan HSV Menggunakan Metode KNN," *Jurnal SAINTEKOM*, vol. 13, no. 2, pp. 214–221, Sep. 2023, doi: 10.33020/saintekom.v13i2.420.
- [2] J. Khatib Sulaiman, N. Amelia, M. Garonga, J. Rusman, and I. Artikel Abstrak, "Penerapan Metode K-Nearest Neighbor (Knn) Untuk Klasifikasi Kematangan Buah Kopi," *Indonesian Journal of Computer Science Attribution*, vol. 12, no. 2, pp. 2023–693.
- [3] A. Octaviani, D. Sandya Prasvita, K. Rizki, T. Zulkarnain, and S. Hinggih, "Klasifikasi Tingkat Kematangan pada Buah Rambutan Berdasarkan Fitur Warna Menggunakan KNN dan Ekstraksi Warna HSV." 2021. [Online]. Available: <https://www.kaggle.com/aelchimminut/fruits262>.
- [4] D. Imantata Muhammad and N. Falih, "Penggunaan K-Nearest Neighbor (KNN) untuk Mengklasifikasi Citra Belimbing Berdasarkan Fitur Warna," no. 1, 2021.
- [5] G. M. Tsani, Y. Rahmawati, O. D. Sanyoto, and S. Agustin, "PENGKLASIFIKASIAN DAUN SIRIH DENGAN METODE K-NEAREST NEIGHBOR (KNN) BERBASIS FITUR WARNA GUNA Mendukung Pemanfaatan Tanaman Obat," 2024.
- [6] D. Hardiyanto, D. Anggun Sartika, J. Teknik Elektro, F. Teknologi Industri, and I. AKPRIND Yogyakarta, "EKSTRAKSI FITUR CITRA API BERBASIS EKSTRAKSI WARNA PADA RUANG WARNA HSV dan RGB."
- [7] R. Kosasih et al., "Classification of Banana Ripe Level Based on Texture Features and KNN Algorithms," 2021.
- [8] A. Syarif and A. Ramadhanu, "KLASIFIKASI CITRA TINGKAT KEMATANGAN BUAH ALPUKAT BERDASARKAN BENTUK WARNA MENGGUNAKAN METODE K-NEAREST NEIGHBO," 2024. [Online]. Available: <http://jurnal.goretanpena.com/index.php/JSSR>
- [9] Y. Reswan, R. Toyib, H. Witriyono, and A. Anggraini, "Klasifikasi Tingkat Kematangan Buah Nanas Berdasarkan Fitur Warna Menggunakan Metode K-Nearest Neighbor (KNN)," *Jurnal Media Infotama*, vol. 20, no. 1, 2024.
- [10] P. Kopi, M. Bandung, M. Farhan, and H. Dermawan, "Penerapan Image Processing untuk Mengetahui Tingkat Kematangan Kopi Menggunakan Algoritma K-Nearest Neighbor (KNN) pada".
- [11] S. Raysyah, V. Arinal, and D. I. Mulyana, "KLASIFIKASI TINGKAT KEMATANGAN BUAH KOPI BERDASARKAN DETEKSI WARNA MENGGUNAKAN METODE KNN DAN PCA," *Sistem Informasi |*, vol. 8, no. 2, pp. 88–95, 2021.
- [12] Melladia; Indah Febri Annisa, "Perancangan Sistem Penanganan Penyakit Tanaman Padi Menggunakan Metode Case Based Reasoning," *J. Sains Teknol. dan Sist. Inf.*, vol. 4, no. 1, pp. 50–57, 2024, doi: 10.54259/satesi.v4i1.2948.
- [13] S. Suparno and A. Rufaidah, "Analisis Perbandingan Metode Moving Average dan Exponential Smoothing untuk Meramalkan Permintaan Produk Turning Pada CV. Gavra Perkasa," *J. Optim.*, vol. 7, no. 2, 2021, doi: 10.35308/jopt.v7i2.4311.
- [14] M. Melladia, Indah Febri Annisa, and Surya Kharisma Karnefo, "Perancangan Data Mart Analisis Data Nilai Siswa pada SMAN 2 Tebo," *JUMINTAL J. Manaj. Inform. dan Bisnis Digit.*, vol. 3, no. 1, pp. 18–30, 2024, doi: 10.55123/jumintal.v3i1.3501.

- [15] A. F. Hastawan, R. Septiana, Y. E. Windarto,
“Perbaikan Hasil Segmentasi HSV Pada Citra
Digital Menggunakan Metode Segmentasi RGB
Grayscale” Edu Komputika Jurnal, 6 (1) 2019