

**Klasifikasi Gambar Landscape Berdasarkan Fitur Warna dan Tekstur Menggunakan Algoritma KNN****Nurul Fuad<sup>1</sup>**<sup>1</sup>Program Studi Ilmu Komputer Fakultas Teknik Universitas GresikEmail: <sup>1</sup>nurulfuad23@gmail.com**Abstract**

Content-Based Image Retrieval (CBIR) is a technology that enables image searching based on content. CBIR operates by measuring the similarity between a query image and all images in the database; consequently, the query cost is directly proportional to the number of images in the database. Limiting the search range through image classification is one method to reduce query costs in CBIR. This study aims to implement the K-Nearest Neighbor algorithm for landscape image classification and to measure the resulting accuracy and classification time. Software was developed to extract color and texture features from landscape images using the Color Histogram and Edge Histogram Descriptor methods. The extracted features were then utilized by the software for learning and classification using the K-Nearest Neighbor method. The software was developed using structured analysis and design methods and implemented in MATLAB. The resulting software was tested based on accuracy and classification time. Test results indicate that combining color and texture features yields higher accuracy compared to classification based on color or texture features alone, although it requires a longer classification time.

**Keywords:** *Klasifikasi gambar, ekstraksi fitur, k-nearest neighbor*

**1. Pendahuluan (numbering tidak diaktifkan, style = Heading 1)**

Pencarian gambar dengan menggunakan keyword yang sering memberikan hasil yang ambigu mendorong lahirnya teknologi Content Based Image Retrieval (CBIR). CBIR dapat membantu proses pencarian gambar berdasarkan content gambar tersebut. CBIR bekerja dengan cara menerima masukan query berupa gambar dan membandingkan gambar query dengan semua gambar yang ada di dalam database kemudian menampilkan hasil query secara terurut berdasarkan ukuran kemiripan antara gambar query dengan gambar pada database [8].

Sayangnya, jumlah gambar dalam database yang sangat banyak akan meningkatkan query cost pada CBIR. Salah satu solusinya adalah membatasi range pencarian dengan melakukan klasifikasi gambar. Klasifikasi gambar tidak hanya meningkatkan akurasi, tetapi juga meningkatkan kecepatan query pada image retrieval [12].

Untuk melakukan klasifikasi terhadap gambar, harus ditetapkan terlebih dahulu ciri atau fitur yang digunakan sebagai pembeda antara gambar yang satu dengan gambar yang lain. Salah satu metode yang dapat digunakan untuk mengklasifikasi gambar berdasarkan fitur yang dimilikinya adalah

K-Nearest Neighbor.

Dalam penelitian ini, yang menjadi pokok pembahasan adalah bagaimana menerapkan K-Nearest Neighbor untuk klasifikasi gambar landscape serta bagaimana tingkat akurasi dan waktu klasifikasi dari K-Nearest Neighbor classifier tersebut. Agar penelitian tetap terfokus maka fitur yang akan diteliti adalah warna dan tekstur, metode ekstraksi fitur warna yang akan digunakan adalah Color Histogram dan untuk mengekstrak fitur tekstur digunakan metode Edge Histogram Descriptor.

(1)

## 2. Metode

### 2.1 Klasifikasi

Klasifikasi adalah proses mencari sekumpulan model atau fungsi yang mendeskripsikan dan membedakan kelas-kelas data untuk tujuan agar fungsi tersebut dapat digunakan untuk memprediksi label objek yang label kelasnya tidak diketahui. Model yang dihasilkan adalah berdasarkan hasil analisis terhadap sekumpulan data *learning* [4].

### 2.2 Ekstraksi Fitur

Fitur dari sebuah gambar adalah karakteristik atau atribut dari sebuah gambar yang dapat membedakannya dari gambar yang lain [11].

Pada referensi [5] disebutkan bahwa jika diinginkan sebuah sistem yang dapat membedakan objek-objek dengan tipe yang berbeda, maka pertama-tama harus ditetapkan karakteristik objek yang dapat diukur sebagai parameter yang mendeskripsikan objek. Karakteristik inilah yang disebut dengan fitur.

Dari beberapa definisi diatas maka ekstraksi fitur dapat didefinisikan sebagai proses mengekstrak karakteristik deskriptif dari sebuah gambar dengan menggunakan metode tertentu.

Fitur yang dipilih harus mampu mendeskripsikan objek yang diwakilinya dengan baik. Adapun kriteria fitur yang baik menurut [5] adalah sebagai berikut:

#### 1. *Discrimination*

Fitur diharapkan mampu memberikan perbedaan nilai yang signifikan untuk objek-objek yang tidak berada dalam satu kelas.

#### 2. *Reliability*

Fitur diharapkan mampu memberikan perbedaan nilai yang minimal untuk objek-objek yang berada dalam satu kelas yang sama.

#### 3. *Independence*

Jika digunakan beberapa fitur, sebaiknya fitur-fitur tersebut tidak saling berkorelasi satu sama lain, artinya nilai fitur yang satu tidak mempengaruhi nilai fitur yang lain.

#### 4. *Small Number*

Fitur diharapkan memiliki dimensionalitas yang kecil sehingga jika direpresentasikan sebagai vektor, maka vektor yang dihasilkan akan memiliki jumlah elemen yang sesedikit mungkin.

### 2.3 Prinsip Kerja K-Nearest Neighbor

*K-Nearest Neighbor* adalah *supervised learning algorithm* dimana sebuah objek diklasifikasikan berdasarkan kelas mayoritas dari  $k$  buah tetangga terdekatnya. Klasifikasi memanfaatkan mekanisme *voting* dari  $k$  buah objek terdekat dan bila hasil *voting* seri, maka label untuk objek akan dipilih secara acak. [9]

*K-Nearest Neighbor* berdasarkan konsep '*learning by analogy*'. Data *learning* dideskripsikan dengan atribut numerik  $n$ -dimensi. Tiap data *learning* merepresentasikan sebuah titik dalam ruang  $n$ -dimensi [4].

Jika sebuah data *query* yang labelnya tidak diketahui diinputkan, maka *K-Nearest Neighbor* akan mencari  $k$  buah data *learning* yang jaraknya paling dekat dengan data *query* dalam ruang  $n$ -dimensi. Jarak antara data *query* dengan data *learning* dihitung dengan cara mengukur jarak antara titik yang merepresentasikan data *query* dengan semua titik yang merepresentasikan data *learning* dengan rumus *Euclidean Distance*.

Diberikan 2 buah titik P dan Q dalam sebuah ruang vektor  $n$ -dimensi dengan  $P(p_1, p_2, \dots, p_n)$  dan  $Q(q_1, q_2, \dots, q_n)$ , maka jarak antara P dan Q dapat diukur dengan menggunakan persamaan *Euclidean Distance* sebagai berikut [4]:

$$D_{\text{euc}}(P, Q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (2.4)$$

dimana P dan Q adalah titik pada ruang vektor  $n$  dimensi sedangkan  $p_i$  dan  $q_i$  adalah besaran skalar untuk dimensi ke  $i$  dalam ruang vektor  $n$  dimensi.

$K$  buah data *learning* terdekat akan melakukan *voting* untuk menentukan label mayoritas. Label data *query* akan ditentukan berdasarkan label mayoritas dan jika ada lebih dari satu label mayoritas maka label data *query* dapat dipilih secara acak di antara label-label mayoritas yang ada.[9]

## 2.4 Akurasi Sistem

Tingkat akurasi sebuah *classifier* untuk sekelompok data uji adalah rasio perbandingan jumlah data uji yang dapat diklasifikasikan dengan benar dengan jumlah seluruh data uji. [4]. Dengan demikian, tingkat akurasi  $K$ -Nearest Neighbor untuk sekelompok data uji  $T$  dapat dinyatakan dalam persamaan berikut:

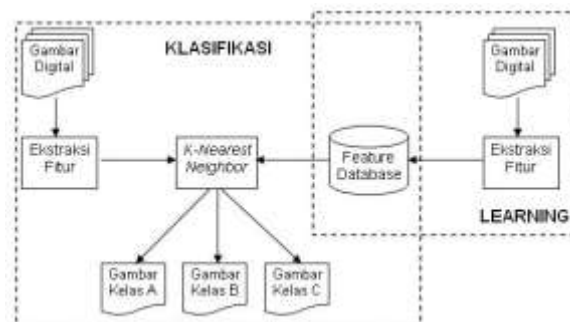
$$Acc(T) = \frac{T_{pos}}{n} \quad (2.5)$$

dimana  $T_{pos}$  adalah jumlah data uji yang diklasifikasikan dengan benar dan  $n$  adalah jumlah data uji seluruhnya.

Dengan demikian tingkat akurasi maksimum = 1 bila semua data uji dapat diklasifikasikan dengan benar dan tingkat akurasi minimum = 0 jika tidak satupun data uji dapat diklasifikasikan dengan benar. Akurasi dapat dinyatakan dalam satuan persen dengan mengalikan hasil perhitungan dengan 100%.

Bagian ini berisi metode yang digunakan oleh penulis dalam menyelesaikan permasalahan yang diteliti.

## 3. Hasil dan Pembahasan



Gambar 3-1: Framework Sistem

Perangkat lunak akan terdiri dari 2 buah proses utama yang saling berkaitan, yaitu proses *learning* dan proses klasifikasi.

Input untuk proses *learning* adalah kumpulan *learning image* yang sudah diketahui label kelasnya. Adapun output yang dihasilkan adalah fitur gambar yang disimpan dalam sebuah *feature database*.

Input untuk proses klasifikasi adalah gambar yang akan diklasifikasikan dan fitur *learning image* yang tersimpan dalam *feature database*. Adapun outputnya adalah label kelas gambar yang diinputkan.

Dengan demikian jelaslah bahwa proses *learning* harus dilakukan sebelum proses klasifikasi sebab output dari proses *learning* menjadi salah satu input yang dibutuhkan dalam proses klasifikasi.

## 2.5 Langkah-langkah Tahap Learning

Pada proses *learning*, input perangkat lunak adalah kumpulan *learning image* yang telah diketahui label kelasnya. Semua *learning image* akan diekstrak fiturnya dengan menggunakan metode *Color Histogram* dan *Edge Histogram Descriptor*. Hasil dari proses ekstraksi fitur akan disimpan dalam sebuah *feature database*.

Berikut ini adalah langkah-langkah yang dilakukan dalam proses *learning*:

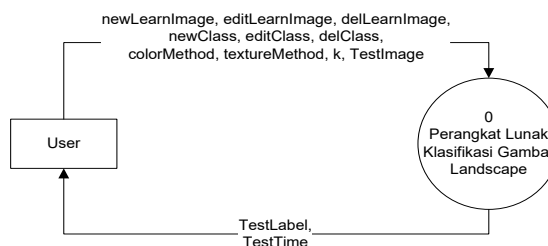
1. Membaca data bitmap *learning image*
2. Melakukan ekstraksi fitur warna dengan metode Color Histogram
3. Melakukan ekstraksi fitur tekstur dengan metode Edge Histogram Descriptor

## 2.6 Perancangan Sistem

Perancangan perangkat lunak dilakukan dengan metode terstruktur. Adapun model-model yang dihasilkan adalah sebagai berikut:

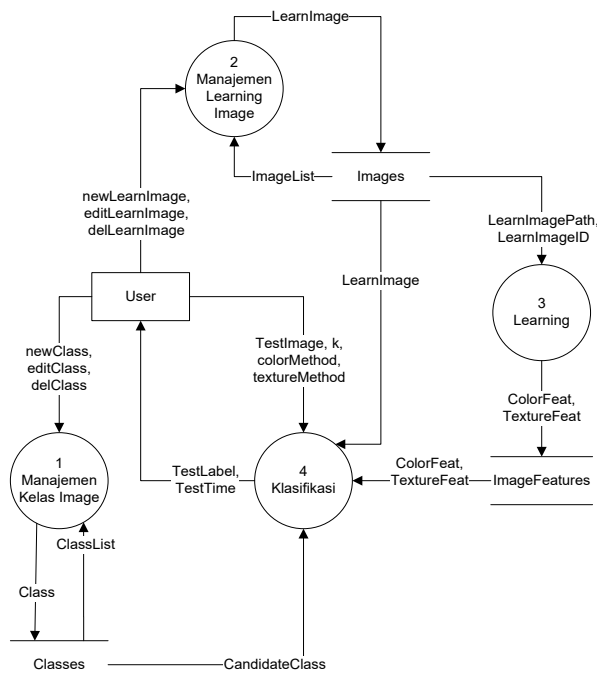
1. Diagram Konteks
2. *Data Flow Diagram* (DFD)
3. Spesifikasi Proses
4. Kamus Data.

Berikut ini adalah diagram konteks dan *data flow diagram* dari sistem yang telah dibangun:



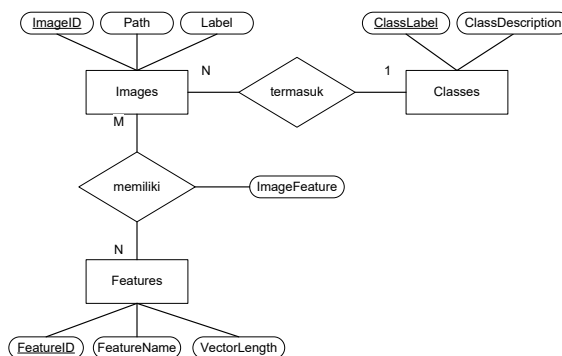
Gambar 3-2: Diagram Konteks

Diagram konteks hanya menggambarkan input dan output sistem secara keseluruhan. Adapun proses-proses inti yang ada di dalam sistem serta aliran data yang terjadi antar proses digambarkan dalam *data flow diagram* berikut ini:



Gambar 3-3: DFD Level 1

Adapun untuk keperluan penyimpanan data, maka diperlukan sebuah basis data. Rancangan basis data dapat dilihat pada diagram Entity-Relationship berikut ini:



Gambar 3-4 Entity Relationship Diagram

Adapun hasil implementasi dari rancangan basis data pada diagram ER di atas dapat dilihat melalui struktur tabel berikut ini:

Tabel 3-1: Struktur tabel 'Classes'

| No | Field            | Type Data | Keterangan                    |
|----|------------------|-----------|-------------------------------|
| 1  | ClassLabel       | String    | Primary key, label dari kelas |
| 2  | ClassDescription | String    | Deskripsi kelas               |

Tabel 3-2: Struktur tabel 'Features'

| No | Field        | Type Data | Keterangan           |
|----|--------------|-----------|----------------------|
| 1  | FeatureID    | Number    | Primary key          |
| 2  | FeatureName  | String    | Nama fitur           |
| 3  | VectorLength | Number    | Jumlah elemen vektor |

Tabel 3-3: Struktur tabel 'Images'

| No | Field   | Type Data | Keterangan                                                                                             |
|----|---------|-----------|--------------------------------------------------------------------------------------------------------|
| 1  | ImageID | Number    | Primary key                                                                                            |
| 2  | Label   | String    | Label dari <i>learning image</i> , foreign key yang mengacu ke field 'ClassLabel' pada tabel 'Classes' |
| 3  | Path    | String    | Path absolute ke file learning image                                                                   |

Tabel 3-4: Struktur tabel 'ImageFeatures'

| No | Field        | Type Data | Keterangan                                                         |
|----|--------------|-----------|--------------------------------------------------------------------|
| 1  | ImageID      | Number    | Foreign key yang mengacu ke field 'ImageID' pada tabel 'Images'    |
| 2  | FeatureID    | Number    | Foreign key yang mengacu ke field 'FeatureID' pada tabel 'Feature' |
| 3  | ImageFeature | String    | Fitur yang diekstrak dari image dan disimpan dalam bentuk string.  |

## 2.7 Pengujian

Data yang digunakan untuk pengujian berasal dari *Best Photo Collection CD* yang terdiri dari 200 gambar yang terbagi dalam 4 kelas yaitu *canyon*, *mountain*, *sunset* dan *waterfall*.

Tabel 3-1: Data untuk pelatihan dan pengujian

| Kelas        | Jumlah data learning | Jumlah data testing |
|--------------|----------------------|---------------------|
| Canyon       | 25 gambar            | 25 gambar           |
| Mountain     | 25 gambar            | 25 gambar           |
| Sunset       | 25 gambar            | 25 gambar           |
| Waterfall    | 25 gambar            | 25 gambar           |
| <b>Total</b> | <b>100 gambar</b>    | <b>100 gambar</b>   |

Parameter yang diuji adalah tingkat akurasi dan waktu klasifikasi. Tingkat akurasi dihitung dengan menghitung persentase gambar yang dapat diklasifikasikan dengan benar dalam satuan persen sedangkan waktu klasifikasi dihitung mulai proses pembacaan data bitmap sampai menghasilkan label kelas dan diukur dalam satuan detik per gambar.

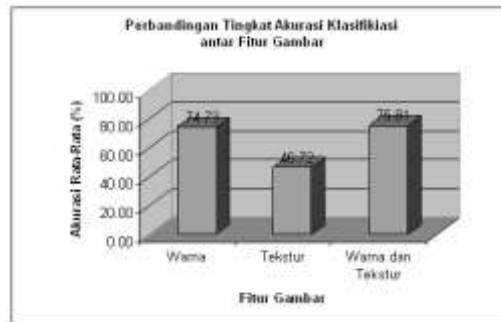
Total jumlah percobaan yang dilakukan adalah 375 percobaan yang terdiri dari:

1. 75 percobaan klasifikasi berdasarkan fitur warna yang diekstrak dengan metode *Color Histogram* 256, 4 dan 32 bin untuk nilai  $k$  masing-masing 1 sampai 25.
2. 75 percobaan klasifikasi berdasarkan fitur tekstur yang diekstrak dengan metode *Edge Histogram Descriptor* 4x4, 3x3 dan 2x2 region untuk nilai  $k$  masing-masing 1 sampai 25.
3. 225 percobaan klasifikasi berdasarkan fitur warna dan tekstur yang diekstrak dengan 9 kombinasi metode ekstraksi fitur warna dan tekstur untuk nilai  $k$  masing-masing 1 sampai 25.

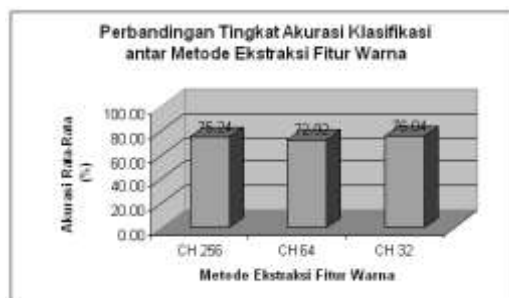
Tabel 4-2: Skenario Pengujian

| Variabel Klasifikasi          |                                                     |             | Jumlah Percobaan |
|-------------------------------|-----------------------------------------------------|-------------|------------------|
| Fitur                         | Metode Ekstraksi Fitur                              | Nilai $k$   |                  |
| Warna                         | Color Histogram 256 bin                             | 1 sampai 25 | 25               |
|                               | Color Histogram 64 bin                              | 1 sampai 25 | 25               |
|                               | Color Histogram 32 bin                              | 1 sampai 25 | 25               |
| Tekstur                       | Edge Histogram 4x4 Region                           | 1 sampai 25 | 25               |
|                               | Edge Histogram 3x3 Region                           | 1 sampai 25 | 25               |
|                               | Edge Histogram 2x2 Region                           | 1 sampai 25 | 25               |
| Warna dan Tekstur             | Color Histogram 256 bin + Edge Histogram 4x4 Region | 1 sampai 25 | 25               |
|                               | Color Histogram 256 bin + Edge Histogram 3x3 Region | 1 sampai 25 | 25               |
|                               | Color Histogram 256 bin + Edge Histogram 2x2 Region | 1 sampai 25 | 25               |
|                               | Color Histogram 64 bin + Edge Histogram 4x4 Region  | 1 sampai 25 | 25               |
|                               | Color Histogram 64 bin + Edge Histogram 3x3 Region  | 1 sampai 25 | 25               |
|                               | Color Histogram 64 bin + Edge Histogram 2x2 Region  | 1 sampai 25 | 25               |
|                               | Color Histogram 32 bin + Edge Histogram 4x4 Region  | 1 sampai 25 | 25               |
|                               | Color Histogram 32 bin + Edge Histogram 3x3 Region  | 1 sampai 25 | 25               |
|                               | Color Histogram 32 bin + Edge Histogram 2x2 Region  | 1 sampai 25 | 25               |
| <b>Total Jumlah Percobaan</b> |                                                     |             | <b>375</b>       |

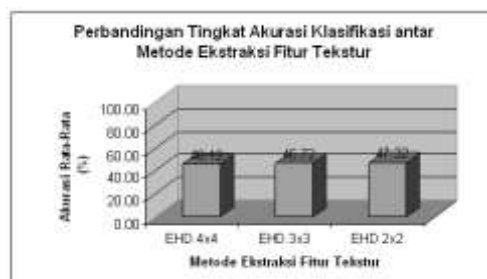
Dalam tiap percobaan digunakan 100 gambar testing. Adapun hasil pengujian dapat dilihat dalam grafik berikut:



Gambar 4-1: Grafik perbandingan tingkat akurasi klasifikasi antar fitur  
Fitur gambar yang digunakan dalam klasifikasi akan mempengaruhi tingkat akurasi klasifikasi. Klasifikasi dengan berdasarkan fitur warna dan tekstur memiliki tingkat akurasi rata-rata yang paling tinggi yaitu sebesar 75,81%.

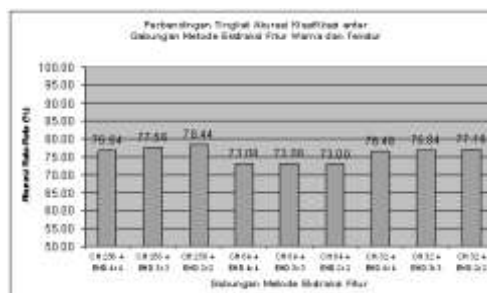


Gambar 4-2: Grafik perbandingan tingkat akurasi klasifikasi antar metode ekstraksi fitur warna  
Metode ekstraksi fitur warna berpengaruh terhadap tingkat akurasi klasifikasi. Adapun metode ekstraksi fitur warna yang memberikan tingkat akurasi klasifikasi tertinggi adalah *Color Histogram 32 bin* (CH 32) yaitu sebesar 76,04 %.



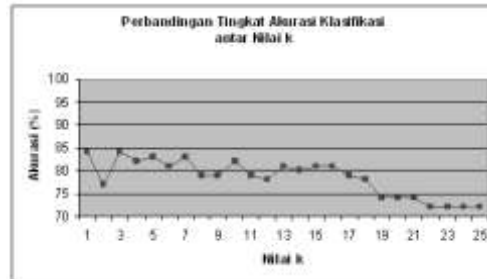
Gambar 4-3: Grafik perbandingan tingkat akurasi kaslifikasi antar metode ekstraksi fitur

Metode ekstraksi fitur tekstur berpengaruh terhadap tingkat akurasi klasifikasi. Adapun metode ekstraksi fitur tekstur yang memberikan tingkat akurasi klasifikasi tertinggi adalah *Edge Histogram Descriptor 2x2 Region* (EHD 2x2) yaitu sebesar 47,32 %.



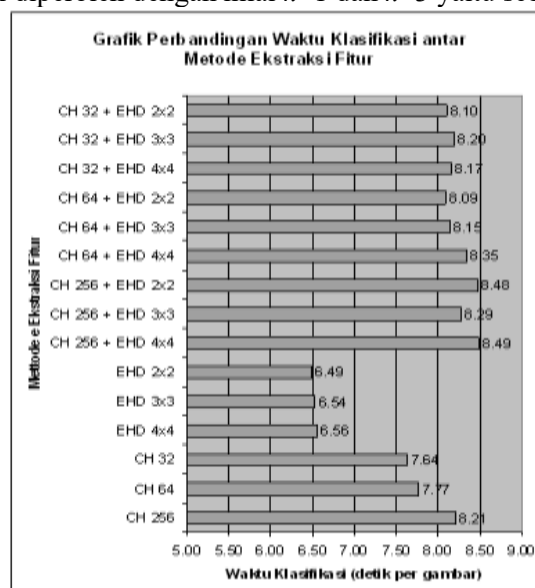
Gambar 4-4: Grafik perbandingan tingkat akurasi klasifikasi antar gabungan metode ekstraksi fitur warna dan tekstur

Dari grafik di atas terlihat bahwa gabungan metode ekstraksi fitur warna dan tekstur yang memberikan tingkat akurasi klasifikasi paling tinggi adalah *Color Histogram 256 bin (CH 256)* dan *Edge Histogram Descriptor 2x2 Region (EHD 2x2)*. Gabungan kedua metode ini memberikan tingkat akurasi rata-rata sebesar 78,44%.



Gambar 4-5: Grafik perbandingan tingkat akurasi klasifikasi antar nilai k

Grafik di atas menunjukkan bahwa tingkat akurasi klasifikasi berfluktuasi tergantung nilai  $k$  yang diinputkan namun cenderung menurun seiring naiknya nilai  $k$ . Untuk klasifikasi berdasarkan fitur warna dengan metode *ColorHistogram 256 bin* dan fitur tekstur dengan metode *Edge histogram 2x2 Region*, tingkat akurasi paling tinggi diperoleh dengan nilai  $k=1$  dan  $k=3$  yaitu sebesar 84 %.



Gambar 4-6: Grafik perbandingan waktu klasifikasi

Waktu klasifikasi tersingkat adalah klasifikasi berdasarkan fitur tekstur yang diekstrak dengan metode *Edge Histogram Descriptor 2x2 Region (EHD 2x2)* yaitu selama 6.49 detik per gambar sedangkan yang membutuhkan waktu paling lama adalah klasifikasi berdasarkan warna dan tekstur yang diekstrak dengan metode *Color Histogram 256 bin* dan *Edge Histogram Descriptor 4x4 Region (CH 256 + EHD 4x4)* yaitu selama 8.49 detik per gambar.

#### 4. Kesimpulan

Dari hasil pembangunan sistem ini serta dari hasil uji coba yang telah dilakukan, dapat ditarik kesimpulan bahwa fitur gambar, metode ekstraksi fitur dan jumlah neighbor yang melakukan voting mempengaruhi tingkat akurasi klasifikasi gambar dengan metode *K-Nearest Neighbor*.

Fitur yang memberikan tingkat akurasi paling tinggi adalah fitur warna dan tekstur sedangkan metode ekstraksi fitur yang memberikan tingkat akurasi tertinggi adalah kombinasi *Color Histogram 256 bin* dan *Edge Histogram Descriptor 2x2*. Tingkat akurasi klasifikasi berfluktuasi tergantung jumlah *neighbor* yang melakukan *voting* namun cenderung menurun seiring bertambahnya jumlah *neighbor* yang melakukan *voting*.

Adapun klasifikasi yang membutuhkan waktu paling singkat adalah klasifikasi dengan fitur tekstur, disusul kemudian oleh klasifikasi dengan fitur warna dan yang membutuhkan waktu paling lama adalah klasifikasi berdasarkan fitur warna dan tekstur.

#### Referensi

- [1] Halder, R. K., et al. (2024). Enhancing K-Nearest Neighbor Algorithm: A Comprehensive Review and Performance Analysis of Modifications. *Journal of Big Data*.
- [2] Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing* (4th Edition)
- [3] Jiawei Han, Micheline Kamber, 2002, "Data Mining Concept and Techniques", Academic Press
- [4] Kenneth R. Castleman, 1996, "Digital Image Processing", Prentice Hall
- [5] Maher A. Sid Ahmed, 1995, "Image Processing: Theory, Algorithm and Architecture", McGrawHill
- [6] Rafael C. Gonzales, Richard E. Woods, 2002, "Digital Image Processing", Pentice Hall Sundaram RMD, "Image Mining, Intricacies and Innovations", <http://www.amrita.edu/cde/>, didownload pada tanggal 11 April 2007
- [7] Teknomo, Kardi, K-Nearest Neighbors Tutorial, 2006, <http://people.revoledu.com/kardi/tutorial/KNN/> didownload pada tanggal 11 Desember 2006
- [8] Uniform Quantization, 2007, <http://www.cs.wpi.edu/~matt/courses/cs563/>, didownload pada tanggal 11 April 2007