

PENERAPAN EKSTRAKSI CIRI TRANSFORMASI WAVELET DALAM PEMBUATAN MODEL KLASIFIKASI KESEGARAN IKAN SELAR

Salsabilah Khansa¹, Jayanta², Ika Nurlaili Isnainiyah³
Program Studi Informatika / Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia
salsabilahkhansa99@gmail.com, anta.jayanta@gmail.com, nurlailika@upnvj.ac.id

Abstrak. Angka konsumsi ikan nasional di Indonesia pada tahun 2019 sebesar 55,95 kg/kapita yang mana melampaui target yang ditetapkan. Selanjutnya untuk tahun 2020, KKP menargetkan angka konsumsi ikan nasional menjadi 56,39 kg/kapita. Salah satu ikan yang persebarannya cukup luas dan hampir bisa ditemukan di seluruh perairan Indonesia adalah ikan selar. Pedagang maupun pembeli sering kali ingin mengetahui apakah kualitas ikan yang dijual maupun dibeli masih dapat dikatakan baik dan masih dapat disimpan di dalam suhu dingin atau tidak. Salah satu cara untuk mengetahui kondisi ikan jika dilihat dari bidang keilmuan Informatika adalah dengan membuat model yang mengekstraksi ciri objek melalui citra dengan serangkaian metode pengolahan citra. Ekstraksi ciri citra ikan menggunakan metode Transformasi *Wavelet* dan untuk membantu klasifikasi kesegaran ikan menggunakan algoritma *K-Nearest Neighbor* (KNN). Model berhasil melakukan klasifikasi citra ikan dengan rata-rata akurasi paling baik pada Level Dekomposisi 2 sebesar 80% tanpa menggunakan *Cross Validation* dan 87,66% menggunakan *Cross Validation*.

Kata Kunci: *image processing, discrete wavelet transform, k-nearest neighbor* dan ikan selar.

1 Pendahuluan

Saat ini, kesadaran masyarakat Indonesia untuk mengonsumsi ikan memang masih cukup rendah dibandingkan dengan negara lain di Asia. Pada tahun 2019, Direktorat Jenderal PDSPKP (Penguatan Daya Saing Produk Kelautan dan Perikanan) yang merupakan bagian dari Kementerian Kelautan dan Perikanan (KKP) melakukan survei penghitungan sementara mengenai angka konsumsi ikan nasional di 34 provinsi, yang ternyata mencapai sebesar 55,95 kg/kapita/tahun [1]. Angka tersebut sudah melampaui target nasional tahun 2019 dan meningkat dari tahun 2018 yang sebesar 50,69 kg/kapita/tahun [2]. Selanjutnya untuk tahun 2020, KKP menargetkan angka konsumsi ikan nasional meningkat menjadi 56,39 kg/kapita/tahun.

Salah satu ikan yang persebarannya cukup luas dan hampir bisa ditemukan di seluruh perairan Indonesia adalah ikan selar. Menurut Genisa, wilayah persebaran ikan selar terdapat di hampir seluruh daerah pantai Indonesia, Teluk Benggala dan beberapa daerah lainnya di luar Indonesia [3]. Berdasarkan data tentang konsumsi menurut jenis makanan di daerah perkotaan dan pedesaan, pada tahun 2017 ikan selar (segar dan diawetkan) memiliki tingkat partisipasi sebesar 8,87, lalu pada tahun 2018 sebesar 8,90, dan pada tahun 2019 sebesar 9,57 [4]. Ikan selar kuning dapat dikategorikan sebagai bagian dari jenis ikan ekonomis yang dikonsumsi oleh masyarakat. Ikan ini sering diolah menjadi pindang, ikan bakar, maupun ikan asin oleh masyarakat [5]. Salah satu cara untuk mengetahui kondisi ikan jika dilihat dari bidang keilmuan Informatika adalah dengan mengekstraksi ciri objek melalui citra dengan metode *image processing*. Dengan memanfaatkan metode tersebut pedagang ikan dapat mengetahui apakah kualitas ikan yang dijual masih dapat dikatakan baik dan masih dapat disimpan di dalam suhu dingin (disimpan di dalam lemari pendingin atau disimpan menggunakan es batu) atau tidak.

Pada penelitian tahun 2016 yang dilakukan oleh Bee, Winsy, dan Altien [6], menggunakan mata ikan selar sebagai objek, dengan metode kuadrat kecil berdasarkan nilai RGB dan *grayscale*. Hasil penelitian menunjukkan bahwa

akurasi system sebesar 83,33%. Kemudian penelitian tahun 2017 yang dilakukan oleh Iswari, Wella, dan Ranny [7] membandingkan 3 jenis algoritma klasifikasi yaitu, KNN, C4.5, dan Naïve Bayes dalam melakukan klasifikasi kesegaran ikan. Hasil penelitian menyatakan bahwa KNN memiliki akurasi paling tinggi, yaitu dengan rata rata 91,359%. Yang terakhir, penelitian tahun 2019 yang dilakukan oleh Miftahur dan Budi [8] mengidentifikasi tingkat kesegaran ikan kembung melalui citra insangnya dengan metode wavelet diskrit haar dan k-means clustering. Hasil penelitian hanya menyatakan bahwa system berhasil mengidentifikasi kesegaran ikan. Pada penelitian ini penulis akan mencoba membuat model yang dapat melakukan klasifikasi kesegaran ikan melalui *image processing*. *Image processing* terdiri dari beberapa tahap yang mana setiap tahap memiliki metode tersendiri. Ekstraksi ciri citra ikan menggunakan metode Transformasi *Wavelet* lalu untuk membantu klasifikasi kesegaran ikan menggunakan algoritma *K-Nearest Neighbor* (KNN). Terdapat 2 (dua) kelas, yaitu kelas dengan kondisi ikan yang masih dapat disimpan di dalam suhu dingin dan kelas dengan kondisi ikan yang sudah tidak dapat disimpan di dalam suhu dingin.

2 Dasar Teori

2.1 Citra Digital

Citra digital terbentuk dari sinyal diskrit dan merupakan citra yang bisa ditampilkan serta dapat diolah oleh komputer. Citra digital dibentuk oleh kumpulan *pixel* berukuran $N \times M$ di mana N adalah baris dan M adalah kolom. Menurut Ramadhan dan Setiyono, citra digital merupakan fungsi dua dimensi yang dapat disimbolkan dengan $f(x,y)$, di mana x dan y merupakan titik koordinat spasial yang berfungsi sebagai koordinat posisi setiap *pixel* [8]. Sedangkan amplitudo dari $f(x,y)$ adalah nilai intensitas cahaya (*brightness*) atau tingkat keabuan. Dapat disebut citra digital jika keseluruhan dari x , y , dan amplitudo f nilainya berhingga (*finite*) dan diskrit [9].

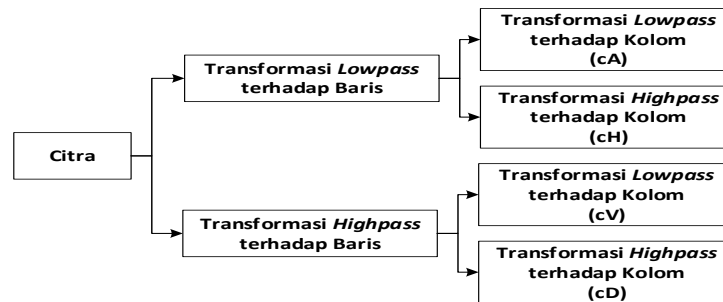
2.2 Pengolahan Citra Digital

Secara umum, pengolahan citra digital adalah langkah-langkah yang dilakukan untuk melakukan proses terhadap citra 2 dimensi [9]. Algoritma untuk mengenali pola kemiripan antara satu citra dengan citra yang lain yaitu meliputi tiga elemen dasar, yaitu pemahaman data, ekstraksi ciri, dan klasifikasi [10]. Lebih lanjut, tahapan pengolahan citra digital yang akan dipakai adalah sebagai berikut [10],

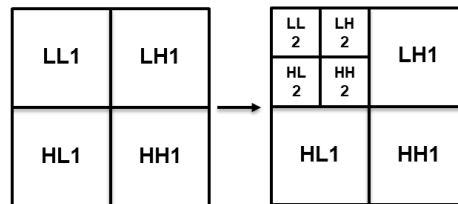
1. Akuisisi citra yaitu menangkap citra analog menggunakan kamera digital dan menjadikannya data citra digital berupa nilai *red*, *green*, *blue* yang memiliki rentang 0 sampai 255.
2. *Image preprocessing* yang di dalam prosesnya terdapat *image enhancement*, yaitu mempersiapkan atau memperbaiki citra agar sesuai dengan kebutuhan ekstraksi fitur.
3. *Image segmentation* yaitu pemisahan objek gambar (*foreground*) dengan latar gambar (*background*).
4. Seleksi dan ekstraksi ciri yaitu pengambilan ciri pada objek citra digital yang bisa dipakai sebagai pembeda antara suatu objek dengan objek-objek lain yang sejenis.
5. *Classifying* atau *Clustering* yaitu algoritma yang mampu menentukan objek masuk ke dalam kelas atau kelompok berdasarkan karakteristik ciri yang dimiliki. Objek akan dimasukkan ke dalam kelas atau kelompok yang anggotanya memiliki kesamaan ciri dengan objek tersebut.
6. Penyusunan basis pengetahuan yaitu memasukkan hasil proses pelabelan (klasifikasi atau pengelompokan) ke dalam referensi sebagai pengenalan objek selanjutnya.

2.3 Transformasi Wavelet

Secara konseptual, proses Transformasi *Wavelet* pada citra cukup sederhana. Transformasi *Wavelet* membagi citra awal atau citra asli (didekomposisi) menjadi 4 *sub-image*. Setiap *sub-image* memiliki ukuran $\frac{1}{4}$ (satu per empat) kali dari citra asli. Tiga *sub-image* yang berada di bagian atas kanan, bawah kiri, dan bawah kanan terlihat seperti citra asli dengan versi tekstur yang lebih kasar (*rough*). Ini terjadi karena bagian-bagian tersebut berisi elemen frekuensi tinggi dari citra asli. Sebaliknya, 1 *sub-image* yang terletak di bagian sudut atas kiri terlihat seperti citra asli yang terlihat lebih halus karena berisi elemen frekuensi rendah dari citra asli. *Sub-image* yang ada pada posisi sudut atas kiri ini dapat dibagi lagi menjadi 4 *sub-image*, seperti langkah awal. Proses tersebut dapat dilakukan berulang, sesuai dengan tingkatan (*level*) transformasi yang diinginkan [9].



Gambar. 113. Proses Transformasi Citra dengan Level Transformasi 1



Gambar. 114. Gambaran Transformasi Wavelet 2 Level

2.4 Nilai Fitur yang Dipakai

Peneliti menggunakan beberapa nilai yang dijadikan sebagai *predictor* untuk menentukan jarak ketetanggaan antara satu citra dengan citra yang lain. Setiap bagian hasil transformasi, dicari nilai *mean*, *mean absolute deviation* dan *median absolute deviation*. Tidak semua nilai penghitungan akan dipakai. Dalam memilih nilai fitur, peneliti menggunakan cara manual Pemilihan nilai fitur berdasarkan persebaran data per kelas di setiap fitur. Apabila persebaran data antara fitur terlalu bercampur, maka fitur tersebut tidak dipilih karena kemungkinan besar akan ada cukup banyak data yang *missclass*. Sedangkan jika dalam suatu fitur cukup terlihat bahwa dua kelas terpisah, kecil kemungkinan untuk data akan *missclass*.

2.4.1 Mean

Mean merupakan nilai rata-rata suatu kelompok data. Pada citra, *mean* dapat diartikan sebagai jumlah nilai seluruh piksel dibagi dengan jumlah elemen piksel. Rumus menghitung mean (\bar{x}) adalah sebagai berikut,

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \tag{1}$$

2.4.2 Mean dan Median Absolute Deviation (MAD)

Fungsi *mad* di Matlab memiliki arti untuk menghitung *mean absolute deviation* atau *median absolute deviation*. *Mean absolute deviation* merupakan jarak rata-rata antara setiap titik data dengan rata-rata. Rumus menghitung *mean absolute deviation* adalah sebagai berikut,

$$MAD = \frac{\sum |x_i - \bar{x}|}{n}$$

di mana
 $\bar{x} = \text{mean}(x)$

(2)

Sedangkan *median absolute deviation* merupakan simpangan absolut dari median data. Metode ini juga cukup *robust* untuk mengukur sebaran data. Rumus menghitung *median absolute deviation* adalah sebagai berikut,

$$MAD = \text{median}(|x_i - \bar{x}|)$$

di mana
 $\bar{x} = \text{median}(x)$

(3)

2.5 K-Nearest Neighbor (KNN)

Menurut Iswari, Wella dan Ramsy, Algoritma KNN merupakan salah satu metode pengenalan pola yang memiliki akurasi tinggi. Algoritma KNN termasuk *supervised algorithm* yaitu sebuah penghampiran di mana sudah terdapat data yang dilatih, lalu menargetkan suatu objek tertentu yang berdasarkan *variable* untuk masuk ke dalam kelompok yang berisi kumpulan data yang telah memiliki label [7].

Algoritma *K-Nearest Neighbor* :

1. Tentukan nilai dari parameter K (jumlah tetangga paling dekat), berupa bilangan bulat positif.
2. Menghitung *Euclidean Distance* antara objek dengan *data training* yang sudah ada. Berikut merupakan formula dari *Euclidean Distance*,

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$
(4)

3. Mengurutkan hasil pada langkah nomor 2 secara *ascending* (berurutan dari nilai yang paling rendah ke nilai yang paling tinggi).
4. Menentukan kelas dengan klasifikasi *nearest neighbor* berdasarkan nilai ketetanggaan K dan nilai *Euclidean Distance* yang paling kecil.
5. Dengan menggunakan label *data training* yang paling mayoritas berdasarkan langkah kedua sampai kelima maka kelas objek akan dapat diprediksi.

2.6 Library untuk Membuat Model : Matrix Laboratory (MATLAB)

Menurut Sianipar, MATLAB merupakan suatu perangkat yang padu yang dapat menunjang pemakainya untuk melakukan komputasi matematik, menganalisa data, mengembangkan algoritma, melakukan simulasi dan pemodelan, serta menghasilkan tampilan grafik dan *user interface* sederhana [11].

2.7 Ikan

2.7.1 Ikan Selar

Selar kuning (*Selaroides leptolepis*) atau yang biasa disebut selar, merupakan jenis ikan pelagis kecil yang merupakan perenang cepat dan kuat, serta pemakan plankton. Ikan ini biasanya berukuran kurang dari 15 cm namun dapat mencapai 22 cm. Bentuknya lonjong memanjang dan pipih tegak dengan ciri khas punggung biru metalik serta terdapat garis kuning membujur dari bagian atas mata ke belakang tubuh hingga ke batang ekor. Ciri lainnya adalah terdapat satu noktah hitam menonjol di bagian atas penutup insang. Gizi yang terkandung dalam ikan selar hampir sama seperti ikan-ikan lainnya. Secara komposisi kimiawi, ikan selar terdiri dari 75,3% — 76,0% air, 17,7% — 21,0% protein, dan 1,9% — 4,6% lemak [12].



Gambar. 115. Ikan Selar 116 mm

2.7.2 Kualitas Ikan

Menurut Hadiwiyoto, secara umum, kualitas ikan yang baik adalah ikan yang masih segar [12]. Tentu saja kondisi ini dapat diperoleh dari hasil penanganan dan sanitasi yang baik. Keadaan kesegaran ikan yang paling maksimal ketika ikan baru saja ditangkap dan diangkat dari dalam air. Semakin lama ikan terpapar udara terbuka, maka semakin menurun pula kesegarannya. Pada umumnya, perubahan fisik ikan dapat dilihat dalam setiap fase. Pada fase 1, kulit ikan terlihat cerah berkilau, berair dan berlendir bening. Pada fase 2, kulit ikan masih terlihat cerah tetapi sudah tidak berkilau dan lendir yang mulai keruh. Pada fase 3, kulit ikan berubah berubah warna dan kusam serta lendirnya seperti susu. Yang terakhir pada fase 4, kulit ikan menjadi pucat dan kusam (sedang dalam kondisi pembusukan yang lebih lanjut) serta berlendir buram.

Selain itu, salah satu bagian dari *sensory methods* penilaian kualitas ikan adalah *demerit scores* di mana minimal skor 0 (kualitas ikan dapat diterima) dan maksimal skor 20 (kualitas ikan ditolak). Ada korelasi linier antara *sensory quality* yang dinyatakan sebagai skor *demerit* dengan masa penyimpanan di atas es. Diketahui bahwa maksimal penyimpanan ikan di atas es adalah 10 (sepuluh) hari karena pada umumnya saat hari ke-10 tersebut, secara *sensory*, ikan sudah mencapai angka maksimal dari skor *demerit* [13]. Penelitian ini akan menggunakan deskripsi di atas sebagai acuan menentukan apakah ikan masuk ke dalam kelas yang masih bisa disimpan di atas es atau tidak. Parameter ikan secara *sensory* atau fisikawi ini digunakan karena kondisi-kondisinya yang dapat diperhatikan oleh mata. Alasan lainnya adalah dengan parameter fisikawi, dapat mengetahui kondisi ikan tanpa merusak tubuh ikan.

3 Perancangan dan Implementasi Model

3.1 Pengumpulan Data Citra

Pada penelitian ini, data citra didapatkan dengan cara mengambil citra permukaan kulit ikan selar. Terdapat 15 ekor ikan selar yang diamati dan diambil citra di kedua sisinya selama 12 (dua belas) hari. Selama kurun waktu tersebut, ikan hanya disimpan di lemari es bagian *freezer* dengan suhu di bawah 0°C. Kemudian setiap hari pada jam yang sama, ikan dikeluarkan dari *freezer* dan didiamkan dalam suhu ruangan selama ±1 jam agar es yang

menempel pada kulit ikan mencair. Selanjutnya, ikan tersebut diambil citranya menggunakan kamera DSLR dengan jarak 15 cm dari objek. Gambar. 4. dan Gambar. 5. merupakan salah satu contoh citra ikan dari masing-masing kelas.



Gambar. 4. Hasil Pengambilan Citra Ikan Selar Kelas Storable



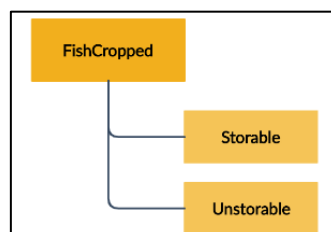
Gambar. 5. Hasil Pengambilan Citra Ikan Selar Kelas Unstorable

Selama 12 (dua belas) hari, terkumpul data sebanyak 288 citra. Namun karena terdapat dua kelas yang rentang waktunya berbeda yaitu, kelas pertama 10 (sepuluh) hari sedangkan kelas kedua hanya 2 (dua) hari. Maka dalam penelitian ini, diputuskan untuk hanya mengambil citra di satu sisi ikan pada 10 (sepuluh) hari pertama, 6 (enam) citra per hari. Sedangkan 2 (dua) hari terakhir mengambil citra di kedua sisi ikan, 30 (tiga puluh) citra per hari. Sehingga jumlah data citra ikan yang siap dipakai adalah 120 citra. Perbandingan yang dipakai untuk membagi data latih dan data uji adalah 80:20.

Kode yang digunakan untuk membaca file citra, menjadikannya sebuah *image datastore* lalu membagi data citra tersebut seperti yang direncanakan adalah sebagai berikut,

```
imageFolder = 'MATLAB\FishCropped';  
imds = imageDatastore(imageFolder, 'LabelSource', 'foldernames', 'IncludeSubfolders',true);  
[trainingSet, testSet] = splitEachLabel(imds, 0.8);
```

Label dapat langsung dideteksi dari nama *sub folder* yang ada di dalam *folder* 'FishCropped'. Visualisasi struktur *folder* dapat dilihat pada Gambar. 6.



Gambar. 6. Struktur Folder

Pembagian jumlah data citra yang digunakan menjadi data latih dan data uji dapat dilihat pada Tabel 1.

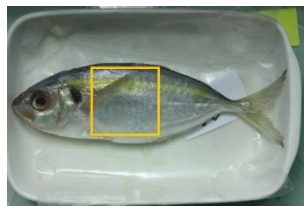
Tabel 1. Jumlah Data Citra Latih dan Data Citra Uji

	<i>Layak</i>										<i>Tidak Layak</i>		<i>Jumlah</i>	
	<i>Hari ke-</i>													
	1	2	3	4	5	6	7	8	9	10	11	12		
<i>Data Latih</i>					48								48	96
<i>Data Uji</i>					12								12	48
<i>Jumlah</i>					60								60	120

3.2 Preprocessing

Setelah data terkumpul, tahapan selanjutnya adalah *preprocessing*. Tahapan ini bertujuan untuk menyiapkan citra agar ciri citra yang dimaksud lebih mudah didapat pada saat tahap *feature extraction*. *Preprocessing* citra yang dilakukan adalah *cropping*, *resizing*, dan *equalizing histogram*.

3.2.1 Cropping



Gambar. 7. Citra Ikan Sebelum Dipotong (Crop)

Resolusi citra yang didapatkan dari hasil pengambilan data citra adalah 5184×3456 pixel seperti pada **Gambar 4.4**. Lalu dilakukan pemotongan (*crop*) citra dengan fitur *imcrop* dari MATLAB dengan beberapa *variable* yaitu, $[x, y, width, height]$. Dalam penelitian ini, *variable* tersebut disesuaikan dengan bagian citra yang ingin diambil. Citra hasil pemotongan berukuran 1000×1000 pixel seperti pada Gambar. 8



Gambar. 8. Citra Ikan Kelas *Storable* Setelah Dipotong (*Crop*)

Kode yang digunakan untuk mendapatkan ukuran tersebut adalah sebagai berikut,

```
croppedImage = imcrop(image,[2000 1200 1000 1000]);
```

3.2.2 *Resizing*

Citra berukuran 1000 x 1000 *pixel* yang merupakan hasil proses *cropping* masuk ke tahap *preprocessing* selanjutnya yaitu *resizing*. *Resizing* merupakan kegiatan mengubah dimensi citra namun tetap mempertahankan seluruh gambarnya, biasanya memengaruhi ukuran file citra yang juga otomatis memengaruhi kualitas gambar. Namun, tidak mengapa, sebab memang itu tujuannya, file berukuran kecil dapat mempercepat proses komputasi. Citra hasil *resizing* dapat dilihat pada Gambar. 9.



Gambar. 9. Citra Ikan Setelah *Resizing*

Fungsi MATLAB yang dapat menjalankan proses *resizing* ini adalah “*imresize*” dengan *scale* 0.1 yang berarti ukuran citra akan menjadi 10% dari ukuran citra awal. Awalnya citra berukuran 1000 × 1000 *pixel*, setelah dilakukan perubahan ukuran, ukuran citra menjadi 100 × 100 *pixel*. Kode yang digunakan untuk mendapatkan ukuran tersebut adalah sebagai berikut,

```
resizedImage = imresize(image,0.1);
```

3.2.3 *Equalizing Histogram*

Setelah melalui tahap *resizing*, citra harus melalui tahap *histogram equalization*. Bagaimanapun penerangan yang direncanakan agar objek memiliki cahaya yang cukup, terkadang masih ada beberapa gambar yang memiliki histogram yang terkonsentrasi pada nilai intensitas tertentu. Tujuan dari tahapan ini adalah untuk meningkatkan kontras gambar dengan meratakan histogram. Citra hasil *histogram equalization* dapat dilihat pada Gambar. 10.



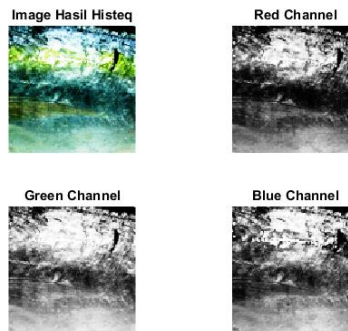
Gambar. 10. Citra Ikan Setelah *Equalizing Histogram*

Fungsi MATLAB yang dapat menjalankan proses ini secara otomatis adalah “*histeq*”. Kode yang digunakan untuk mendapatkan ukuran tersebut adalah sebagai berikut,

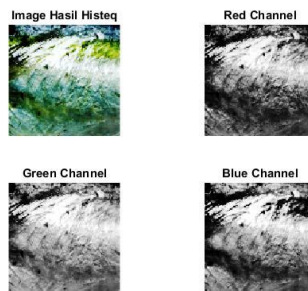
```
equalizedHistImage = histeq(image);
```


3.2.4 *Extracting Blue Channel*

Setelah citra memiliki kontras yang cukup, citra masuk ke dalam tahap pengambilan *blue channel*. *Channel* yang dimaksud di sini adalah versi *grayscale* dari citra asli yang hanya terbuat dari salah satu warna primer. Misalnya dalam warna *RGB* terdiri dari tiga *channels* yaitu, *red*, *green*, dan *blue*. Beberapa objek dapat terlihat lebih cerah di *channel* tertentu, tergantung pada warna objek itu sendiri. Dilihat setiap *channel*nya pada citra *RGB* kulit ikan. Terlihat pada Gambar. 11. dan Gambar. 12. bahwa pada *blue channel*, sedikit garis kuning pada badan ikan lebih terlihat dibandingkan di dua *channel* lainnya. Oleh sebab itu, yang dipakai dalam *preprocessing* ini adalah *blue channel*.



Gambar. 11. Hasil Ekstraksi RGB Channel Kelas *Storable*



Gambar. 12. Hasil Ekstraksi RGB Channel Kelas *Unstorable*

Kode yang digunakan untuk mendapatkan *blue channel* dari sebuah citra yang sudah melalui tahapan pra proses ekualisasi histogram sebelumnya adalah sebagai berikut,

```
blueChannel = equalizedHistImage (:, :, 3);
```

3.3 Feature Extraction

Citra yang sudah melewati tahap *preprocessing* dianggap sudah siap untuk masuk ke dalam *feature extraction*. Pada tahap *feature extraction* menggunakan metode Transformasi *Wavelet* dengan *filter* Haar. Fitur MATLAB yang dapat digunakan untuk melakukan proses tersebut terhadap citra adalah “*haart2*” yang berarti transformasi *wavelet* dua dimensi dengan *filter* Haar. *Level* dekomposisi yang digunakan pada penelitian ini adalah sampai *level* 5 karena angka tersebut merupakan *level* maksimal untuk ukuran *image* 100 × 100 pixel. Kode yang digunakan adalah sebagai berikut,

```
[a,h,v,d] = haart2(image,Level,'integer');
```

Lalu pada *level* tersebut, setiap bagian (*approximation*, *horizontal detail*, *vertical detail*, dan *diagonal detail*) dicari nilai *Mean*, *Mean Absolute Deviation*, dan *Median Absolute Deviation (MAD)*. Rumus menghitung ketiga nilai tersebut ada pada persamaan 1, persamaan 2, dan persamaan 3.

Fungsi MATLAB yang dapat menjalankan perhitungan *Mean* adalah “*mean*” sedangkan untuk menjalankan perhitungan *Mean* dan *Median Absolute Deviation* “*mad*”. Kode yang digunakan untuk menghitung nilai *Mean* dan *Median Absolute Deviation* adalah sebagai berikut,

```
meanA = mean(waveletA(:));
```

```
MADA = mad(waveletA(:));
```

Sebenarnya tahapan *feature extraction* ini menghasilkan 8 (delapan) fitur atau *variable*. Namun setelah melakukan uji coba dengan menghilangkan beberapa fitur, ternyata dapat menghasilkan akurasi yang lebih baik. Pada penelitian ini diputuskan untuk menggunakan 5 fitur yaitu, *mean* dari *approximation* (a) dan *vertical detail* (v), lalu *median absolute deviation* dari *horizontal* (h), *vertical* (v), dan *diagonal detail* (d). Lalu langkah terakhir dari proses ini adalah menormalisasi nilai tersebut dengan *range* 0—1. Contoh nilai fitur yang dihasilkan ditampilkan pada Tabel 2.

Tabel 2. Nilai Fitur Citra

File Name	MeanA	MeanV	MADH	MADV	MADD	Label
Image58	0,624851	0,756857	0,592674	0,856782	0,843828	Storable
Image57	0,499298	0,893460	0,606955	0,873296	0,982733	Unstorable

3.4 Perancangan Model

Untuk mengetahui *level* dekomposisi dan nilai ketetanggaan yang menghasilkan akurasi paling baik, nilai-nilai tersebut harus di uji coba terlebih dahulu. Data yang dipakai untuk uji coba perancangan model adalah data latih. Terdapat lima nilai *level* dekomposisi : 1, 2, 3, 4, dan 5. Sedangkan terdapat 5 nilai ketetanggaan : 1, 3, 5, 7, dan 9. Sepuluh nilai tersebut akan menghasilkan 25 kombinasi. Kode yang digunakan untuk klasifikasi data dengan K-NN adalah sebagai berikut,

```
M = fitcknn(TrainTable,'Label','NumNeighbors',K);
```

4 Hasil dan Pembahasan

4.1 Pelatihan Model

Klasifikasi data menggunakan algoritma *K-Nearest Neighbor* berdasarkan fitur yang telah didapatkan pada tahap *feature extraction*. Pelatihan model ini menggunakan *Level* Dekomposisi = 2. Cara kerja algoritma *K-Nearest Neighbor* dalam mengklasifikasikan data latih yang ada dalam penelitian ini adalah sebagai berikut,

1. 80% data merupakan data latih bukan dipilih secara acak, melainkan 80% data pertama dari setiap kelas. Sehingga data latih berisi 48 citra kelas “Storable” dan 48 citra kelas “Unstorable”.
2. Mencari jarak atau selisih nilai antar citra yang diwakili oleh 5 variable (*mean approximation*, *mean vertical detail*, *MAD horizontal detail*, *MAD vertical detail*, dan *MAD diagonal detail*). Jenis jarak yang dipakai adalah *Euclidean Distance* dengan nilai K atau ketetanggaan yang dipakai adalah 1,3,5,7, dan 9.
3. Semakin dekat jarak citra dengan citra lain maka model akan menganggap citra tersebut memiliki kemiripan dan dimasukkan ke dalam kelas yang sama.

4.2 Performa Model

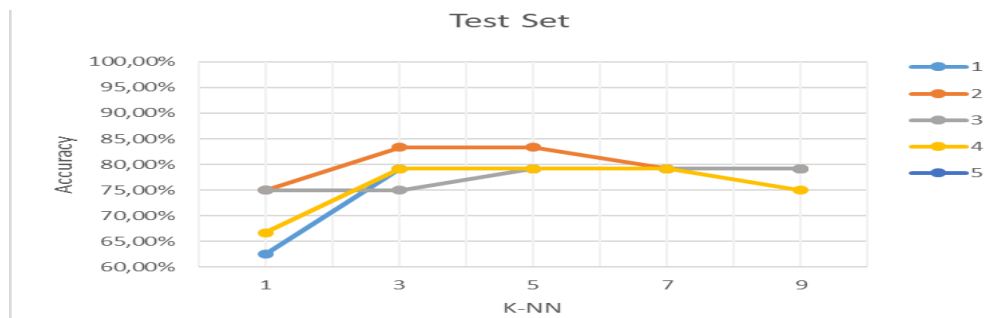
Tahap pelatihan model menggunakan data latih. Diketahui bahwa dengan *Level* Dekomposisi = 2 mendapatkan rata-rata akurasi yang paling baik yaitu 95,21% dan 90,62% pada data latih. Maka pada penerapan model terhadap data uji ini akan menggunakan skenario tersebut. Dengan asumsi bahwa jika pada data latih memiliki akurasi yang baik, seharusnya pada data uji juga memiliki akurasi yang baik. Data uji yang dipakai berjumlah 24 data citra yang terdiri dari 12 citra yang memiliki kelas asli “Storable” dan 12 citra yang memiliki kelas asli “Unstorable”.

4.1.1 Uji Model Tanpa *K-Fold Cross-Validation*

Akurasi dari kombinasi KNN dan level dekomposisi data uji yang tanpa menggunakan *K Fold Cross-Validation* ada pada Tabel 3. dan disajikan dalam diagram garis pada Gambar. 13.

Tabel 3. Kombinasi KNN dan Level Dekomposisi Data Uji Tanpa Cross-Validation

Accuracy	Decomposition Level					Average
	1	2	3	4	5	
1	62,50%	75,00%	75,00%	66,67%	50,00%	65,83%
3	79,17%	83,33%	75,00%	79,17%	41,67%	71,67%
K 5	79,17%	83,33%	79,17%	79,17%	45,83%	73,33%
7	79,17%	79,17%	79,17%	79,17%	37,50%	70,84%
9	79,17%	79,17%	79,17%	75,00%	54,17%	73,34%
Average	75,84%	80,00%	77,50%	75,83%	45,83%	71,00%



Gambar. 13. Grafik Akurasi Model KNN dengan Data Uji Tanpa Cross-Validation

Tabel 5. menunjukkan bahwa rata-rata akurasi terbaik ada pada *Level* Dekomposisi = 2 yaitu 80%. Pada *level* dekomposisi tersebut, akurasi paling baik bernilai 83,33% yaitu pada saat $K = 3$ dan $K = 5$.

4.1.2 Uji Model dengan *K-Fold Cross-Validation*

Model juga diuji dengan *K-Fold Cross-Validation* di mana $K = 5$ lipatan (*fold*). Jumlah data latih dan data uji pada setiap *fold* sama seperti jumlah data pada pengujian tanpa *Cross-Validation*. Pembagian data latih dan data uji yang dimaksud adalah seperti yang ada pada Tabel 4.

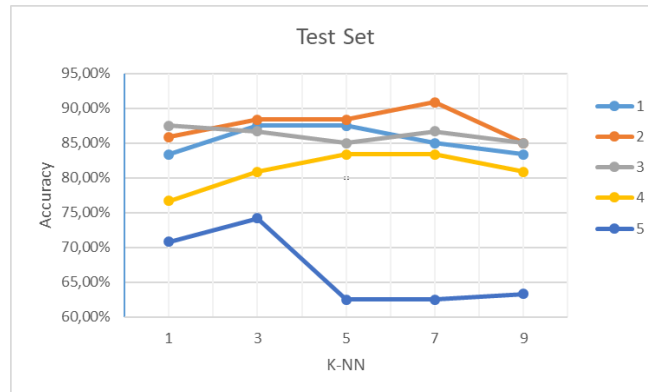
Tabel 4. Jumlah Data Pada Setiap Fold

	Fold				
	1	2	3	4	5
Data Latih	96	96	96	96	96
Data Uji	24	24	24	24	24
Jumlah	120	120	120	120	120

Akurasi dari kombinasi KNN dan level dekomposisi data uji yang menggunakan *K Fold Cross-Validation* ada pada Tabel 5. dan disajikan dalam diagram garis pada Gambar. 14.

Tabel 5. Kombinasi KNN dan Level Dekomposisi Data Uji Dengan Cross-Validation

Accuracy	Decomposition Level					Average
	1	2	3	4	5	
1	83,33%	85,83%	87,50%	76,67%	70,83%	80,83%
3	87,50%	88,33%	86,67%	80,83%	74,17%	83,50%
K 5	87,50%	88,33%	85,00%	83,33%	62,50%	81,33%
7	85,00%	90,83%	86,67%	83,33%	62,50%	81,67%
9	83,33%	85,00%	85,00%	80,83%	63,33%	79,50%
Average	85,33%	87,66%	86,17%	81,00%	66,67%	81,37%



Gambar. 14. Grafik Akurasi Model KNN dengan Data Uji Dengan Cross-Validation

Tabel 5. menunjukkan bahwa rata-rata akurasi terbaik ada pada *Level* Dekomposisi = 2 yaitu 87,66%. Pada *level* dekomposisi tersebut, akurasi paling baik bernilai 90,83% yaitu pada saat $K = 7$. Akurasi pada Tabel 5. dapat berubah karena *Cross-Validation* mengambil data latih dan data uji secara acak pada setiap iterasinya. Namun, biasanya hasil akurasi setiap iterasi tidak jauh berbeda

5 Penutup

5.1. Kesimpulan dan Saran

Berdasarkan hasil analisis mengenai penerapan Transformasi *Wavelet* dan *K-Nearest Neighbor* dalam mengekstraksi dan mengklasifikasikan citra kulit ikan, dapat diambil kesimpulan bahwa Metode Transformasi *Wavelet* mampu mengekstraksi ciri dari citra kulit ikan. Ciri yang dihasilkan cukup membantu algoritma *K Nearest Neighbor* dalam mengklasifikasi citra. Akurasi paling tinggi pada data latih didapatkan dengan skenario tanpa menggunakan *K-Fold Cross-Validation* yaitu 100% dengan $K = 1$ pada semua *level* dekomposisi. Sedangkan akurasi paling tinggi dengan skenario menggunakan *K-Fold Cross-Validation* adalah 93,75% pada $K = 1$ dan *Level* Dekomposisi = 2. Rata-rata akurasi paling tinggi pada data uji didapatkan dengan skenario menggunakan *K-Fold Cross-Validation* yaitu 87,66% pada *Level* Dekomposisi = 2. Sedangkan rata-rata akurasi dengan skenario tanpa menggunakan *K-Fold Cross-Validation* adalah 80% pada *Level* Dekomposisi = 2. Untuk mencapai akurasi paling baik di setiap *data set* dibutuhkan pengaturan yang berbeda-beda, bergantung pada karakteristik *data set* itu sendiri. Dalam hal ini pengaturan tersebut adalah *level* dekomposisi, pemilihan ciri/parameter, dan nilai K .

Berdasarkan kesimpulan mengenai penerapan Transformasi *Wavelet* dan *K-Nearest Neighbor* dalam mengekstraksi dan mengklasifikasikan citra kulit ikan, beberapa saran yang dapat digunakan pada penelitian berikutnya, yaitu menambahkan data citra kulit ikan, baik sebagai data latih maupun sebagai data uji agar akurasi meningkat lebih baik dari akurasi di dalam penelitian ini. Selain itu, penelitian berikutnya dapat menggunakan metode ekstraksi ciri citra dan metode klasifikasi lain sebagai perbandingan untuk mencari metode yang paling tepat. Selanjutnya pada kesegaran ikan dapat menggunakan kelas lain yang memiliki karakteristik kelas yang berbeda. Yang terakhir adalah meningkatkan kemampuan model dengan dapat mengidentifikasi objek lain yang bukan ikan. Misalnya dengan memberi informasi kepada *user* bahwa objek tersebut bukan ikan sehingga model tidak dapat melakukan klasifikasi.

Referensi

- [1] Ekarina. Konsumsi Ikan 2019 Diprediksi Capai 54,49 Kg per Kapita [Internet]. 2018. Available from: <https://katadata.co.id/berita/2018/12/17/konsumsi-ikan-2019-diprediksi-capai-5449-kg-per-kapita>
- [2] Desfika TS. KKP Komitmen Tingkatkan Konsumsi Ikan Nasional [Internet]. 2019. Available from: <https://www.beritasatu.com/ekonomi/590319-kkp-komitmen-tingkatkan-konsumsi-ikan-nasional>
- [3] Genisa AS. 1999 - PENGENALAN JENIS - JENIS IKAN LAUT EKONOMI PENTING DI INDONESIA.pdf. 1999. p. 17–38.
- [4] Supriyanto S. Pengeluaran untuk Konsumsi Penduduk Indonesia, Maret 2019. Jakarta: BPS RI; 2019.
- [5] Sharfina, Maizan, Boer M, Ernawati Y. Potensi Lestari Ikan Selar Kuning (Selaroides Leptolepis) di Perairan Selat Sunda. *Mar Fish*. 2014;5(1):101–8.
- [6] Bee D, Weku W, Rindengan A. Aplikasi Penentuan Tingkat Kesegaran Ikan Selar Berbasis Citra Digital Dengan Metode Kuadrat Terkecil. *d’CARTESIAN*. 2016;5(2):121.
- [7] Iswari NMS, Wella W, Ranny R. Perbandingan Algoritma kNN, C4.5, dan Naive Bayes dalam Pengklasifikasian Kesegaran Ikan Menggunakan Media Foto. *J Ultim*. 2017;9(2):114–7.
- [8] Ramadhan MD, Setiyono B. Pengolahan Citra untuk Mengetahui Tingkat Kesegaran Ikan Menggunakan Metode Transformasi Wavelet Diskrit. *J Sains dan Seni ITS*. 2019;8(1):23–8.
- [9] Putra D. Pengolahan Citra Digital. Penerbit Andi; 2010.
- [10] Andono PN, Sutojo T, Muljono. Pengolahan Citra Digital. Penerbit Andi; 2017.
- [11] Sianipar RH. Pemrograman Matlab dalam contoh dan penerapan (Vol. 1). Penerbit Informatika; 2013.
- [12] Hadiwiyoto S. Teknologi Pengolahan Hasil Perikanan. Liberty; 1993.
- [13] Huss HH. Quality and Quality Changes in Fresh Fish [Internet]. 1995 [cited 2020 Mar 1]. Available from: <http://www.fao.org/3/v7180e/v7180e09.htm>