

**KLASIFIKASI KESEGRAN IKAN BERDASARKAN
EKSTRAKSI FITUR MENGGUNAKAN METODE
K-NEAREST NEIGHBOR DAN *HUE SATURATION VALUE***

SKRIPSI

RIZKI NURSAFITRI SITUMORANG

0701163095



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

**KLASIFIKASI KESEGERAN IKAN BERDASARKAN
EKSTRAKSI FITUR MENGGUNAKAN METODE
K-NEAREST NEIGHBOR DAN *HUE SATURATION VALUE***

SKRIPSI

Diajukan untuk Memenuhi Syarat Mencapai Gelar Sarjana Komputer

RIZKI NURSAFITRI SITUMORANG

0701163095



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi
Lamp : -
Kepada Yth.,
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Sumatera Utara Medan

Assalamu'alaikum Wr.Wb.

Setelah membaca, meneliti, memberikan petunjuk, dan mengoreksi serta mengadakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara,

Nama	: Rizki Nursafitri Situmorang
Nomor Induk Mahasiswa	: 0701163095
Program Studi	: Ilmu komputer
Judul	: Klasifikasi Kesegaran Ikan Berdasarkan Ekstraksi Fitur Menggunakan Metode K-Nearest Neighbor dan Hue Saturation Value

Dapat disetujui untuk segera *dimunaqasyahkan*. Atas perhatiannya kami ucapkan terimakasih.

Medan, 10 Agustus 2021 M
1 Muharram 1443 H

Komisi Pembimbing,

Pembimbing Skripsi I,

Pembimbing Skripsi II,

Assalamu'alaikum munaqasah.
25/8/2021
Furqan



Dr. Mhd. Furqan, S.Si, M.Comp.Sc.
NIP. 198008062006041003

Abdul Halim Hasugian, M.Kom.
NIB. 1100000113

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini :

Nama : Rizki Nursafitri Situmorang
Nomor Induk Mahasiswa : 0701163095
Program Studi : Ilmu Komputer
Judul : Klasifikasi Kesegaran Ikan Berdasarkan Ekstraksi Fitur Menggunakan Metode K-Nearest Neighbor dan Hue Saturation Value

Saya menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali ada beberapa kutipan dan ringkasan yang masing-masing diperoleh dari sumbernya sendiri. Apabila di kemudian hari ditemukan plagiarisme dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan, 10 Agustus 2021

Rizki Nursafitri Situmorang
NIM. 0701163095



**KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI SUMATERA
UTARA MEDAN FAKULTAS SAINS DAN
TEKNOLOGI**

Jl. IAIN No.1 Medan 20235

Telp. (061) 6615683-6622925, Fax. (061) 6615683

Url: <http://saintek.uinsu.ac.id>, E-mail: saintek@uinsu.ac.id

PENGESAHAN SKRIPSI

Nomor : B.123/ST/ST.V.2/PP.01.1/06/2022

Judul : Klasifikasi Kesegaran Ikan Berdasarkan Ekstraksi
Fitur Menggunakan Metode K-Nearest Neighbor
dan Hue Saturation Value
Nama : Rizki Nursafitri Situmorang
Nomor Induk Mahasiswa : 0701163095
Program Studi : Ilmu Komputer
Fakultas : Sains dan Teknologi

Telah dipertahankan di hadapan Dewan Penguji Skripsi Program Studi Ilmu Komputer
Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan dinyatakan **LULUS**.

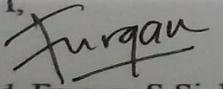
Pada hari/tanggal : Senin, 6 September 2021
Tempat/media : Via Zoom Meeting

Tim Ujian Munaqasyah
Ketua,

Ilka Zufria, M.Kom
NIP. 198506042015031006

Dewan Penguji,

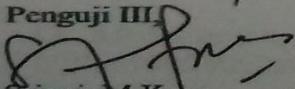
Penguji I,


Dr. Mhd. Furqan, S.Si, M.Comp.Sc.
NIP. 198008062006041003

Penguji II,


Abdul Hafim Hasugian, M.Kom.
NIB. 1100000113

Penguji III,


Sriani, M.Kom.
NIB. 1100000108

Penguji IV,


Rakhmat Kurniawan R, S.T, M.Kom
NIP. 198503162015031003

Mengesahkan,
Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan

Dr. Mhd. Syahnan, M.A
NIP. 196609051991031002

ABSTRAK

Ikan segar merupakan salah satu sumber protein yang tinggi kualitasnya dan sangat baik untuk di konsumsi. Setiap ikan memiliki ciri yang digunakan dalam menentukan kesegaran/kualitasnya, misalnya di lihat dari rupa atau kenampakan, rasa, bau serta tekstur pada ikan. Saat ini, identifikasi kesegaran ikan masih dilakukan secara manual oleh nelayan, penjual, dan juga konsumen. Hal tersebut tentu akan sedikit menyulitkan untuk membedakan antara ikan yang masih dalam kondisi segar maupun ikan yang dalam kondisi tidak segar/mulai membusuk. Salah satu cara untuk mengenali dan mengidentifikasi kesegaran ikan di bidang informatika adalah menggunakan pengolahan citra. Pada penelitian ini akan dibuat suatu sistem pengolahan citra untuk mengklasifikasi kesegaran ikan menggunakan metode *HSV* dan klasifikasi *KNN*. Tahapan analisa yang dilakukan adalah data akuisisi dengan menggunakan kamera smartphone 13 MP dilakukan pemotretan terhadap data ikan untuk dilakukan ekstraksi fitur warna pada mata ikan tersebut sehingga diperoleh citra digital ikan dalam format jpg. Gambar (citra) diambil dari ikan segar dan tidak segar/mulai membusuk. Ekstraksi fitur menggunakan model warna *HSV* untuk fitur warna dan klasifikasi citra menggunakan *KNN* dengan dua tahapan yaitu tahap training dan testing. Data yang digunakan pada penelitian ini adalah 60 citra, yang terdiri dari 46 citra latih dan 14 data uji. Penggunaan *HSV* dan *KNN* telah di uji coba dan mendapatkan hasil akurasi sebesar 93%

Kata Kunci : Kesegaran ikan, *HSV*, Klasifikasi *KNN*

ABSTRACT

Fresh fish is one source of high quality protein and very good for consumption. Each fish has characteristics that are used to determine its freshness/quality, for example in terms of the appearance, taste, smell and texture of the fish. Currently, fish freshness identification is still done manually by fishermen, sellers, and consumers. Of course, it will be a little difficult to distinguish between fish that are still in fresh condition and fish that are not fresh/starting to rot. One way to recognize and identify fish freshness in the field of informatics is to use image processing. In this study, an image processing system will be created to classify fish freshness using the *HSV* method and *KNN* classification. The aim is to determine the accuracy of the method in the classification of fish freshness. The stage of analysis carried out to extract color features on the fish's eye so that digital images of fish are obtained in jpg format. The image (image) is taken from fresh fish and not fresh/starting to rot. Feature extraction using the *HSV* color model for color features and image classification using *KNN* with two stages, namely the training and testing stages. The data used in this study were 60 images, consisting of 46 training images and 14 test data. The use of *HSV* and *KNN* has been tested and obtained an accuracy of 93%

Keywords: Freshness of fish, *HSV*, *KNN* classification

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Syukur Alhamdulillah kehadiran Allah SWT atas rahmatNya yang telah memberikan kemudahan sehingga penulis dapat menyelesaikan skripsi ini dengan judul “Klasifikasi Kesegaran Ikan Berdasarkan Ekstraksi Fitur Menggunakan Metode K-Nearest Neighbor dan Hue Saturation Value (HSV)”.

Penulis menyadari bahwa skripsi ini dapat diselesaikan karena adanya bimbingan, arahan, motivasi, bantuan, do'a serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Syahrin Harahap, M.A, selaku Rektor Universitas Islam Negeri Sumatera Utara Medan.
2. Bapak Dr. Mhd. Syahnan, M.A, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara
3. Bapak Ilka Zufria M.Kom, selaku Ketua Program Studi Ilmu Komputer Fakultas Sains dan Teknologi UIN Sumatera Utara
4. Bapak Dr. Mhd. Furqan, S.Si, M.Comp.Sc selaku Pembimbing Skripsi pertama yang telah banyak membantu serta memberikan bimbingan, motivasi, arahan, ide yang baik serta nasehat selama proses menyelesaikan skripsi ini.
5. Bapak Abdul Halim Hasugian, M.Kom selaku Pembimbing Skripsi kedua yang juga banyak membantu penulis sehingga skripsi ini dapat selesai.
6. Bapak Rakmat Kurniawan R, M.Kom selaku Sekretaris Jurusan Ilmu Komputer, selaku Penguji Skripsi pertama penulis dan sebagai dosen Penasehat Akademik penulis yang juga banyak membantu serta membimbing penulis sejak sah statusnya sebagai mahasiswi ilmu komputer. Semoga senantiasa dimudahkan segala urusannya.
7. Seluruh dosen program studi Ilmu Komputer maupun staff pegawai di Universitas Islam Negeri Sumatera Utara

8. Ibu Sriani, M.Kom selaku Penguji Skripsi kedua saya yang memberikan nasehat, masukan dan arahan sampai dengan selesai
9. Dan yang paling istimewa kedua orangtua penulis, Ayahanda Sahat Situmorang dan Ibunda Rodia Harahap yang tiada hentinya selalu mendoakan, selalu memberikan motivasi, memberikan semangat, dukungan, arahan serta nasehat-nasehat yang baik untuk penulis
10. Kakak dan Abang penulis Nurhasanah Situmorang, Ernita Situmorang dan Ahmad Rasyid Situmorang serta adik penulis Mega Yustika Situmorang, Hotmaruli Tua Situmorang, Nurhamidah Situmorang yang selalu memberikan dukungan kepada penulis.
11. Dan tak lupa untuk yang terkasih, Anugrah Juhriaman yang juga banyak membantu penulis, memberikan semangat dari awal sampai dengan selesainya skripsi ini
12. Teman-teman ilmu komputer 3 serta teman-teman lainnya yang sama-sama sedang berjuang walaupun pada saat itu sedang pandemi, semoga tetap semangat mencapai cita-citanya dan semoga kita selalu di rakhmati Allah SWT.
13. Serta seluruh pihak yang sudah membantu penulis, semoga kebbaikannya di balas oleh Allah SWT.

Penulis menyadari bahwa skripsi ini jauh dari kata sempurna, untuk itu kritik dan saran yang sifatnya membangun sangat penulis harapkan untuk perbaikan dimasa yang akan datang.

Medan, 10 Agustus 2021

Penyusun,

Rizki Nursafitri

0701163095

DAFTAR ISI

	Halaman
ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	vii
DAFTAR TABEL	ix
DAFTAR LAMPIRAN	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Pengertian Ikan Segar dan Tidak Segar	5
2.2 Citra.....	8
2.2.1 Pengertian Citra.....	8
2.2.2 Pengolahan Citra	8
2.3 Jenis-jenis Citra.....	9
2.3.1 Citra Analog	9
2.3.2 Citra Digital.....	10
2.4 Jenis-jenis Citra Digital.....	10
2.4.1 Elemen-elemen Citra Digital	10
2.4.2 Piksel (Picture Element).....	12
2.5 Ekstraksi Fitur / Ciri Warna	13
2.6 Hue Saturation Value (HSV)	13
2.7 K-Nearest Neighbor (K-NN).....	16
2.8 Flowchart	19

2.9 Perhitungan Akurasi.....	21
2.10 Penelitian Terkait	21
BAB III METODE PENELITIAN	23
3.1 Tempat dan Waktu Penelitian	23
3.1.1 Tempat Penelitian.....	23
3.1.2 Waktu Penelitan	23
3.2 Bahan dan Alat Penelitian.....	23
3.2.1 Perangkat Keras	24
3.2.2 Perangkat Lunak.....	24
3.3 Prosedur Kerja.....	24
3.3.1 Teknik Pengumpulan Data.....	25
3.3.2 Analisis kebutuhan	25
3.3.3 Perancangan	25
3.3.4 Pengujian.....	26
3.3.5 Penerapan/Pengujian	27
BAB IV HASIL DAN PEMBAHASAN.....	28
4.1 Pembahasan	28
4.1.1 Analisis Data	28
4.1.2 Representasi Data	29
4.1.3 Hasil Analisis Data.....	36
4.1.4 Perancangan.....	38
4.2 Hasil.....	48
4.2.1 Pengujian	48
4.2.2 Penerapan	65
BAB V KESIMPULAN DAN SARAN	67
5.1 Kesimpulan.....	67
5.2 Saran.....	67
DAFTAR PUSTAKA	68
LAMPIRAN	

DAFTAR GAMBAR

Gambar	Judul Gambar	Halaman
2.1	Proses Pengolahan Citra.....	9
2.2	Ruang Warna HSV pada Matlab	14
2.3	Flowchart Perhitungan K-NN.....	18
3.1	Tahap-tahap Prosedur Kerja.....	24
3.2	Diagram Perencanaan Sistem.....	26
4.1	Citra Ikan Sampel.....	29
4.2	Citra Ikan Uji.....	36
4.3	Rancangan Menu Utama	39
4.4	Tampilan Menu Utama	40
4.5	Rancangan Menu Pelatihan.....	40
4.6	Tampilan Menu Pelatihan.....	41
4.7	Rancangan Menu Pengujian.....	42
4.8	Tampilan Menu Pengujian	43
4.9	Rancangan Menu Tentang Penulis	44
4.10	Tampilan Menu Tentang Penulis	44
4.11	Flowchart Menu Utama	45
4.12	Flowchart Menu Pelatihan	46
4.13	Flowchart Menu Pengujian Air	47
4.14	Flowchart Menu Tentang Penulis.....	48
4.15	Citra Ikan Latih.....	49
4.16	Citra Ikan Uji.....	49
4.17	Menu Proses Pelatihan.....	50
4.18	Menu Proses Pencarian Folder Latih.....	51
4.19	Menu Proses Pelatihan Selesai	51
4.20	Citra Ikan Latih HSV.....	52
4.21	Proses Pengujian Citra Ikan.....	53
4.22	Menu Pencarian Citra.....	53

4.23 Citra Ikan Segar 1 dipilih	54
4.24 Informasi Citra Ikan Segar 1.....	55
4.25 Klasifikasi Citra Ikan Segar 1.....	55
4.26 Klasifikasi Citra Ikan Segar 2.....	56
4.27 Klasifikasi Citra Ikan Segar 3.....	56
4.28 Klasifikasi Citra Ikan Segar 4.....	57
4.29 Klasifikasi Citra Ikan Segar 5.....	57
4.30 Klasifikasi Citra Ikan Segar 6.....	58
4.31 Klasifikasi Citra Ikan Segar 7.....	58
4.32 Klasifikasi Citra Ikan Tidak Segar 1.....	60
4.33 Klasifikasi Citra Ikan Tidak Segar 2.....	60
4.34 Klasifikasi Citra Ikan Tidak Segar 3	61
4.35 Klasifikasi Citra Ikan Tidak Segar 4.....	62
4.36 Klasifikasi Citra Ikan Tidak Segar 5.....	62
4.37 Klasifikasi Citra Ikan Tidak Segar 6.....	63
4.38 Klasifikasi Citra Ikan Tidak Segar 7.....	63

DAFTAR TABEL

Tabel	Judul Tabel	Halaman
2.1	Ciri-ciri Ikan yang segar dan ikan yang kurang segar.....	6
2.2	Simbol-simbol Flowchart.....	20
2.3	Penelitian Terkait.....	21
3.1	Waktu Penelitian.....	23
4.1	Nilai RGB 3x3 Citra Ikan Segar 1	30
4.2	Nilai RGB 3x3 Citra Ikan Segar 2.....	31
4.3	Nilai RGB 3x3 Citra Ikan Tidak Segar 1.....	31
4.4	Nilai RGB 3x3 Citra Ikan Tidak Segar 2.....	32
4.5	Nilai Rata-rata RGB Citra Ikan.....	33
4.6	Nilai HSV Citra Ikan.....	35
4.7	Nilai RGB 3x3 Citra Uji.....	37
4.8	Data Jarak Terdekat Berdasarkan Citra Ikan Uji.....	38
4.9	Informasi Citra Latih dan Citra Uji.....	49
4.10	Hasil Pengujian Citra Ikan Segar.....	59
4.11	Hasil Pengujian Citra Ikan Tidak Segar.....	64
4.12	Hasil Pengujian Klasifikasi Kesegaran Citra Ikan Keseluruhan.....	64

DAFTAR LAMPIRAN

Lampiran	Judul Lampiran
1.	Kode program matlab untuk klasifikasi kesegaran ikan berdasarkan ekstraksi fitur menggunakan metode k-nearest neighbor dan hue saturation value
2.	Hasil uji citra ikan pada Klasifikasi Kesegaran Ikan Berdasarkan Ekstraksi Fitur Menggunakan Metode K-Nearest Neighbor dan Hue Saturation
3.	Kartu Bimbingan Skripsi
4.	Riwayat Hidup

BAB I

PENDAHULUAN

1.1 Latar Belakang

Secara umum ikan merupakan salah satu bahan pangan yang sifatnya tidak tahan lama dikarenakan ikan mengandung air yang cukup tinggi. Oleh karena itu proses penanganannya harus di perhatikan dari saat penangkapan sampai dengan proses dilakukannya penurunan suhu (Winarni dkk., 2003)

Ikan merupakan hewan bertulang belakang (vertebrata) yang berdarah dingin, hidup di air, bergerak dan mempertahankan keseimbangan tubuhnya dengan menggunakan sirip dan bernafas dengan insang, namun selain menggunakan insang ada juga ikan yang memiliki alat pernafasan yang fungsinya sama dengan paru-paru (Irawan, 1995).

Seperti yang kita ketahui sumber protein pada ikan sangat tinggi dan sangat baik untuk di konsumsi terlebih lagi ikan tersebut masih dalam keadaan segar, sebagaimana firman Allah SWT dalam surat Al-Faathir ayat 12

وَمَا يَسْتَوِي الْبَحْرَآ هَذَا عَذْبٌ فُرَاتٌ سَائِغٌ شْرَابُهُ وَهَذَا مِلْحٌ أُجَاجٌ وَمِن كُلِّ تَأْكُلُونَ لَحْمًا
طَرِيًّا وَتَسْتَخْرِجُونَ حِلْيَةً تَلْبَسُونَهَا وَتَرَى الْفُلْكَ فِيهِ مَوَآخِرَ لِيَتَبَتَّعُوا مِن فَضْلِهِ ؕ وَلَعَلَّكُمْ
تَشْكُرُونَ (١٢)

Artinya: “Dan tidak sama (antara) dua lautan; yang ini tawar, segar, sedap diminum dan yang lain asin lagi pahit. Dan dari (masing-masing lautan) itu kamu dapat memakan daging yang segar dan kamu dapat mengeluarkan perhiasan yang kamu pakai, dan di sana kamu melihat kapal-kapal berlayar membelah laut agar kamu dapat mencari karunia-Nya dan agar kamu bersyukur”.

Dari penggalan ayat diatas Allah menganjurkan umatnya untuk mengonsumsi daging yang segar. Namun cara agar kesegarannya tetap terjaga salah satu yang bisa dilakukan yaitu penerapan suhu rendah atau pendinginan menggunakan es. Tingkat kesegaran ikan tersebut juga akan menentukan nilai jual ikan (Surti dan Ari, 2004). Ikan memiliki ciri-ciri untuk melihat kesegaran ataupun kualitasnya, seperti dilihat dari rupa, rasa, bau, maupun tekstur pada ikan. Proses pengenalan pola untuk mengidentifikasi maupun mengklasifikasi dapat dilakukan dengan mengenali ciri-ciri dari sebuah gambar atau citra (Winarni dkk., 2003).

Perkembangan teknologi saat ini sudah sangat berkembang salah satunya ialah tehnik pengolahan citra yang dapat melakukan klasifikasi citra digital. Tahapan dalam proses identifikasi maupun klasifikasi citra digital yaitu akuisisi citra, pra pengolahan citra, ekstraksi ciri/ fitur, pelatihan, pengujian dan pengukuran akurasi. Ekstraksi ciri dalam sebuah citra digital berfungsi untuk mengenali objek yang ada dalam citra tersebut. Ekstraksi ciri ada beberapa jenis yaitu ekstraksi ciri tekstur, ukuran, bentuk, warna maupun geometri. Ekstraksi memerlukan data citra untuk proses pelatihan dan pengujian sistem

Berdasarkan latar belakang yang telah peneliti uraikan, peneliti bermaksud untuk membuat sebuah sistem penelitian yang dapat mengklasifikasi kesegaran ikan menggunakan metode k-nearest neighbor dan metode hue saturation value untuk menganalisis apakah metode tersebut mampu mengklasifikasi kesegaran ikan dengan mendapatkan nilai akurasi yang lebih baik. Hue Saturation Value (HSV) merupakan salah satu metode ekstraksi fitur/ciri warna yang terbukti menjadi deskriptor warna yang bagus dan sudah banyak digunakan. Kemudian menggunakan metode K-Nearest Neighbor sebagai metode pengelompokan atau pengklasifikasiannya. Metode K-Nearest Neighbor merupakan salah satu metode yang bersifat sederhana, dimana metode ini mencari selisih antar vektor ciri dengan menggunakan jarak euclidean untuk menghitung kemiripan 2 vektor pada citra ikan. Penelitian ini diharapkan mampu menghasilkan sistem yang baik untuk mengklasifikasi kesegaran ikan.

1.2 Rumusan Masalah

Rumusan penelitian dari latar belakang tersebut maka muncul permasalahan sebagai berikut :

1. Bagaimana menerapkan metode k-nearest neighbor dan hue saturation value untuk mengidentifikasi kesegaran ikan berdasarkan ekstraksi fitur warna
2. Bagaimana membangun sistem aplikasi dengan menggunakan metode hue saturation value dan k-nearest neighbor untuk klasifikasi kesegaran ikan berdasarkan ekstraksi fitur

1.3 Batasan Masalah

1. Penelitian ini hanya mendeteksi kesegaran berdasarkan ekstraksi fitur warna
2. Metode yang digunakan adalah Hue Saturation Value sebagai ekstraksi warna dan K-Nearest Neighbour sebagai metode klasifikasi
3. Ikan yang digunakan adalah ikan laut
4. Identifikasi dan pengujian dilakukan berdasarkan 2 (dua) kelas yaitu ikan segar dan tidak segar
5. Data yang digunakan sebanyak 60 citra sampel dengan rincian 46 citra latih dan 14 citra uji
6. Citra yang digunakan dengan format jpg/jpeg
7. Tools yang digunakan ialah Matlab

1.4 Tujuan Penelitian

Adapun berdasarkan dari rumusan penelitian tersebut, maka tujuan penelitian ini adalah sebagai berikut :

1. Untuk menerapkan metode hue saturation value dan k-nearest neighbor dalam mengidentifikasi kesegaran ikan berdasarkan ekstraksi fitur warna
2. Untuk membangun sistem aplikasi dengan menggunakan metode hue saturation value dan k-nearest neighbor dalam klasifikasi kesegaran ikan berdasarkan ekstraksi fitur

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah :

1. Dapat mengetahui dan memahami penerapan algoritma k-nearest neighbor dan hue saturation value dalam mengklasifikasi kesegaran pada ikan berbentuk citra digital.
2. Mengetahui bagaimana pencegahan turunnya tingkat kesegaran ikan agar tetap terjaga
3. Mengenali jenis jenis tingkat kesegaran pada ikan dalam bidang image processing.

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Ikan Segar dan Tidak Segar

Secara umum ikan termasuk bahan pangan yang sangat mudah rusak sehingga untuk mempertahankan ataupun menjaga kesegarannya dibutuhkan penanganan segera saat di ambil dari laut hingga sampai ke tangan konsumen dengan memberikan suhu rendah serta memperhatikan kebersihan dan kesehatan yang dapat menentukan nilai jual ikan tersebut. Hal ini dapat dilihat pada ikan-ikan yang baru di tangkap dalam beberapa jam saja kalau tidak diberi perlakuan atau penanganan khusus yang tepat, maka kesegaran maupun mutu ikan tersebut akan menurun (Junianto, 2003). Ikan merupakan hewan vertebrata (bertulang belakang) yang berdarah dingin, hidup di air, bergerak dan mempertahankan keseimbangan tubuhnya dengan menggunakan sirip dan bernafas dengan insang, namun selain menggunakan insang ada juga ikan yang memiliki alat pernafasan yang fungsinya sama dengan paru-paru (Djuhandra, 1998).

Ikan segar atau ikan basah adalah ikan yang belum terdapat proses pengawetan kecuali hanya didinginkan dengan es, dan tentu saja suhu dimana ikan ditempatkan tetap rendah agar tetap stabil. Jenis ikan yang dikatakan segar seperti ikan yang masih hidup dapat dilihat dari rupa, rasa, bau ataupun teksturnya dan tentu saja belum mengalami proses pengolahan lebih lanjut (Adawyah, 2007). Dan sebaliknya ikan yang tidak segar ialah ikan yang sudah mengalami perubahan baik dari rupa, bau, maupun teksturnya. Tingkat kesegaran ikan selanjutnya tentu sangat menentukan peruntukan ikan tersebut untuk proses pengolahan selanjutnya dan sekaligus menentukan nilai jual ikan (Surti dan Ari, 2004). Berikut ini tabel perbedaan ciri-ciri ikan segar dan tidak segar (FAO, 1995) :

Tabel 2.1 Ciri-ciri Ikan Segar dan Tidak Segar / Mulai Membusuk

No	Berdasarkan Jenis Parameter	Kondisi Segar	Kondisi Tidak Segar
1	Mata	Pupil hitam menonjol dengan kornea jernih, bola mata cembung dan cemerlang atau cerah	Pupil mata kelabu tertutup lendir seperti putih susu, bola mata cekung dan keruh
2	Insang	Insang berwarna merah, terang, dan lamella insang terpisah. Insang tertutup oleh lendir berwarna terang dan berbau segar	Insang berwarna coklat, suram atau abu-abu dan lamella insang berdempetan. Lendir insang keruh dan berbau asam, menusuk hidung
3	Sisik	Sisik menempel kuat pada tubuh sehingga sulit dilepas	Sisik mudah terlepas dari tubuh
4	Daging	Daging kenyal, padat, daging melekat pada tulang, berbau segar. Bila daging di tekan tidak ada bekas jari	Daging lunak, mudah lepas dari tulang, berbau busuk. Bila di tekan dengan jari maka bekas tekanannya lama hilang
5	Kulit dan Lendir	Warna kulit terang dan jernih. Kulit masih kuat membungkus tubuh, lendir dipermukaan jernih dan transparan	Berwarna suram, pucat, dan pudar. Kulit mulai terlihat mengendur. Lendir tebal dan menggumpal serta lengket

Tabel Lanjutan 2.1 Ciri – Ciri Ikan Segar dan Tidak Segar / Membusuk

No	Berdasarkan Jenis Parameter	Kondisi Segar	Kondisi Tidak Segar
6	Bau	Berbau khas ikan, berbau laut atau air	Berbau amis yg menyengat
7	Perut dan Sayatan Daging	Perut tidak pecah, masih utuh dan warna sayatan daging cemerlang jika ikan dibelah daging melekat kuat pada tulang terutama rusuknya	Perut sobek, warna sayatan daging kurang cemerlang dan terdapat warna merah sepanjang tulang belakang serta jika dibelah daging mudah lepas

Tingkat kesegaran ikan dapat digolongkan ke dalam empat kelas mutu (Hadiwiyoto, 1993), yaitu :

1. Ikan yang kesegarannya masih baik sekali (prima)

Pada kondisi ini, ikan baru saja ditangkap dan baru saja mengalami kematian. Semua organ tubuh ikan baik daging, mata, maupun insangnya masih benar-benar dalam keadaan segar.

2. Ikan yang kesegarannya masih baik (advanced)

Kondisi pada ikan ini dalam keadaan segar namun tidak sesegar seperti kondisi pertama. Ciri-cirinya adalah bola mata agak cerah, kornea agak keruh, warna insang agak kusam, warna daging masih cemerlang namun lunak bila ditekan.

3. Ikan yang kesegarannya mulai mundur (sedang)

Kondisi pada ikan ini organ tubuhnya sudah banyak mengalami perubahan, bola mata agak cekung, kornea agak keruh, warna insang mulai berubah menjadi merah muda, warna sayatan daging mulai pudar dan daging lembek.

4. Ikan yang sudah tidak segar lagi (busuk)

Ciri-ciri ikan pada kondisi ini adalah daging sudah lunak, sayatan daging sudah tidak cemerlang, bola mata cekung, insang berubah menjadi warna coklat tua, sisik mudah lepas dan sudah menyebarkan bau busuk dan sudah tidak layak lagi lagi dikonsumsi

2.2 Citra

2.2.1 Pengertian Citra

Citra merupakan representasi gambaran, kemiripan dari suatu objek. Citra terbagi ke dalam 2 jenis citra, yaitu citra bersifat analog dan citra bersifat digital. Citra analog seperti gambar pada monitor, televisi, foto dan lain sebagainya. Sedangkan citra digital ialah citra yang dapat diolah kembali oleh komputer (Sutoyo,2009).

Secara harfiah, citra merupakan gambar dua dimensi yang memiliki intensitas cahaya. Sumber cahaya menerangi objek, kemudian objek memantulkan kembali sebagian dari berkas cahaya tersebut. Dari pantulan cahaya tersebut ditangkap oleh alat-alat optik, seperti mata pada manusia, kamera pemindai (scanner) dan lain sebagainya sehingga bayangan dari objek tersebut dapat terekam(Munir, 2004:2).

2.2.2 Pengolahan Citra

Pengolahan citra merupakan usaha yang dilakukan untuk mengubah citra menjadi citra lain yang lebih sempurna sehingga *output* yang dihasilkan sesuai dengan keinginan (Furqan dkk, 2020). Pengolahan citra (image processing) merupakan proses mengolah piksel-piksel di dalam citra digital untuk tujuan tertentu. Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses proses komputer serta munculnya ilmu-ilmu komputasi yang memungkinkan manusia dapat mengambil informasi dari suatu citra (Sutoyo, 2009). Teknik pengolahan citra juga dapat dilakukan dengan mengidentifikasi tekstur pada suatu objek dengan menggunakan metode statistik secara jelas (Furqan dkk, 2020). Proses pengolahan citra secara

diagram, proses dimulai dari pengambilan citra, perbaikan kualitas citra, sampai dengan pernyataan representatif citra yang di citrakan sebagai berikut :



Gambar 2.1 Proses Pengolahan Citra

Dalam perkembangan lebih lanjut, *image processing* dan *computer vision* digunakan sebagai mata manusia, dengan perangkat *input image capture* seperti kamera dan *scanner* dijadikan sebagai mata dan mesin komputer (dengan program komputasinya) dijadikan sebagai otak yang mengolah informasi. Sehingga muncul beberapa pecahan bidang yang menjadi penting dalam *computer vision*, antara lain: *pattern recognition* (pengenalan pola), *biometric* pengenalan identifikasi manusia berdasarkan ciri-ciri biologis yang tampak pada badan manusia, *content based image and video retrieval* (mendapatkan kembali citra atau video dengan informasi tertentu) video editing, dan lain-lain (Basuki, 2005).

2.3 Jenis-Jenis Citra

2.3.1 Citra Analog

Citra analog ialah citra yang tidak dapat direpresentasi kedalam komputer sehingga tidak bisa langsung diproses oleh komputer seperti foto, lukisan, hasil CT scan dan lain sebagainya. Oleh karena itu agar dapat di proses komputer harus di konversi ke bentuk digital. Citra analog dihasilkan dari proses/alat akuisisi, seperti mata manusia dan kamera analog, gambaran yang tertangkap oleh mata manusia yang tertangkap oleh kamera merupakan contoh citra analog. Citra analog memiliki tingkat resolusi yang sangat baik tetapi tidak dapat disimpan, diolah maupun diduplikasi di dalam komputer (Adi P, 2013).

2.3.2 Citra Digital

Pengolahan citra digital merupakan pemrosesan sebuah citra menggunakan komputer agar citra tersebut mudah diinterpretasikan oleh manusia ataupun mesin (Basuki, 2005). Arti dari citra digital sendiri ialah citra yang dapat diproses langsung di dalam komputer, isi dari citra digital adalah nilai kompleks direpresentasikan menggunakan deret bit tertentu (Furqan dkk,2020).

2.4 Jenis-Jenis Citra Digital

Beberapa jenis citra digital yang sering digunakan adalah citra biner, citra grayscale dan citra warna (Sutoyo, 2009).

- a. Citra Biner atau sering disebut citra Monokrom yang memiliki kedalaman bit sebesar 1-bit dan nilai intensitas warna pada setiap piksel citra biner dibagi menjadi 2 warna yaitu warna hitam yang dinyatakan dengan nilai 0 dan warna putih yang dinyatakan dengan nilai 1.
- b. Citra Grayscale atau citra skala keabuan yang hanya memiliki satu kanal warna, kedalaman bit sebesar 8-bit. Setiap piksel memiliki nilai intensitas warna sebanyak 256 derajat warna (0 s/d 255). Nilai 0 menyatakan hitam sempurna dan nilai 255 menyatakan warna putih sempurna. Nilai antara 0 s.d 255 merupakan warna abu-abu.
- c. Citra Warna atau citra true color yang terdiri dari tiga kanal warna yaitu merah, hijau dan biru atau disebut warna *RGB*. citra RGB memiliki kedalaman bit sebesar 24-bit dan memiliki variasi warna sebanyak $256 \times 256 \times 256 = 16.777.216$ pada setiap pikselnya. Itulah alasan mengapa disebut true color karena jumlah warna yang cukup besar dan mencakup semua warna di alam.

2.4.1 Elemen-elemen Citra Digital

Berikut adalah elemen-elemen yang terdapat pada citra digital (Sutoyo, 2009)

1. Kecerahan (*Brightness*).

Brightness merupakan intensitas cahaya yang dipancarkan piksel citra yang dapat ditangkap oleh sistem penglihatan. Kecerahan pada sebuah titik (piksel) di dalam citra merupakan intensitas rata-rata dari suatu area yang melingkupinya.

2. Kontras (*Contrast*).

Kontras menyatakan sebaran terang dan gelap dalam sebuah citra. Pada citra yang baik, komposisi gelap dan terang tersebar secara merata. Citra dengan kontras rendah dicirikan oleh sebagian besar komposisi citranya adalah terang atau sebagian besar gelap. Pada citra dengan kontras yang baik, komposisi gelap dan terang tersebar secara merata.

3. Kontur (*Contour*).

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada piksel-piksel yang bertetangga. Karena adanya perubahan intensitas inilah mata mampu mendeteksi tepi-tepi objek di dalam citra.

4. Warna (*Color*)

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang (λ) yang berbeda. Warna merah mempunyai panjang gelombang paling tinggi, sedangkan warna ungu (*violet*) mempunyai panjang gelombang paling rendah. Warna-warna yang diterima oleh mata (sistem visual manusia) merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda. Penelitian memperlihatkan bahwa kombinasi warna yang memberikan rentang warna yang paling lebar adalah *red* (R), *green* (G), dan *blue* (B). Persepsi sistem visual manusia terhadap warna sangat relatif sebab dipengaruhi oleh banyak kriteria, salah satunya disebabkan oleh adaptasi yang menimbulkan distorsi. Misalnya bercak abu-abu di sekitar warna hijau akan tampak keunguan (distorsi terhadap ruang), atau jika mata melihat warna hijau lalu langsung dengan cepat melihat warna abu-abu, maka mata menangkap kesan warna abu-abu tersebut sebagai warna ungu (distorsi terhadap waktu).

5. Bentuk (*Shape*).

Shape adalah properti intrinsik dari objek 3 dimensi, dengan pengertian bahwa bentuk merupakan properti intrinsik utama untuk sistem visual manusia. Manusia lebih sering mengasosiasikan objek dengan bentuknya ketimbang elemen lainnya (warna misalnya). Pada umumnya, citra yang dibentuk oleh mata merupakan citra dwimatra (2 dimensi), sedangkan objek yang dilihat umumnya berbentuk trimatra (3 dimensi). Informasi bentuk objek dapat diekstraksi dari citra pada permulaan pra-pengolahan dan segmentasi citra. Salah satu tantangan utama pada *computer vision* adalah merepresentasikan bentuk, atau aspek-aspek penting dari bentuk.

6. Tekstur (*Texture*).

Texture dicirikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan piksel-piksel yang bertetangga. Tekstur adalah sifat-sifat atau karakteristik yang dimiliki oleh suatu daerah yang cukup besar, sehingga secara alami sifat-sifat tadi dapat berulang dalam daerah tersebut. Jadi, tekstur tidak dapat didefinisikan untuk sebuah pixel. Sistem visual manusia pada hakikatnya tidak menerima informasi citra secara independen pada setiap *pixel*, melainkan suatu citra dianggap sebagai suatu kesatuan. Resolusi citra yang diamati ditentukan oleh skala pada mana tekstur tersebut dipersepsi. Sebagai contoh, jika kita mengamati citra lantai berubin dari jarak jauh, maka kita mengamati bahwa tekstur terbentuk oleh penempatan ubin-ubin secara keseluruhan, bukan dari persepsi pola di dalam ubin itu sendiri. Tetapi, jika kita mengamati citra yang sama dari jarak yang dekat, maka hanya beberapa ubin yang tampak dalam bidang pengamatan, sehingga kita mempersepsi bahwa tekstur terbentuk oleh penempatan pola-pola rinci yang menyusun tiap ubin.

2.4.2 *Piksel (Picture Element)*

Piksel adalah sebuah citra yang disimpan kedalam sebuah komputer digital harus disimpan ke dalam format yang dapat diolah oleh sebuah program komputer digital. Yaitu dengan cara membagi citra ke dalam sekumpulan sel-sel diskret.

Piksel ini sendiri adalah sebuah kisi-kisi persegi yang kecil. Selanjutnya setiap piksel diberi nilai sesuai dengan nilai kecerahan warna piksel itu sendiri, biasa disebut intensitas piksel. Citra tersusun oleh sekumpulan piksel yang memiliki koordinat (x,y) dan amplitudo $f(x,y)$. Koordinat (x,y) menunjukkan letak/posisi piksel dalam suatu citra, sedangkan amplitudo $f(x,y)$ menunjukkan nilai intensitas warna citra. (Adi P, 2013)

2.5 Ekstraksi Fitur/Ciri Warna

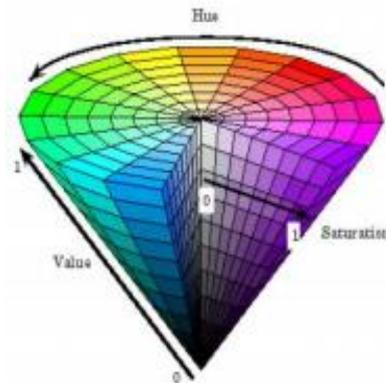
Ekstraksi fitur citra merupakan tahapan mengekstrak ciri/informasi dari objek di dalam citra yang ingin dikenali/dibedakan dengan objek lainnya. Dalam tahap ini terdapat algoritma yang berfungsi untuk encode konten citra menjadi bentuk informasi yang deskriptif. Fitur yang paling relevan pada suatu citra ialah ukuran distribusi (intensitas) warna, tekstur, dan bentuk. Fitur ini biasanya disebut dengan vektor fitur (feature vector) berupa indikator numerik dari konten citra. Fitur tersebut yang akan menjadi modal suatu konten citra dikenali/diklasifikasi (Kadir & Susanto, 2012).

Metode ekstraksi ciri warna yang digunakan dalam penelitian ini adalah metode HSV. Sebelum dilakukan perhitungan nilai H, S, V terlebih dahulu dilakukan konversi citra RGB ke HSV. Konversi RGB ke HSV pada matlab bisa dilakukan dengan menggunakan fungsi 'rgb2hsv'. Pada proses pembuatan histogram HSV, setiap komponen H, S, V (Hue, Saturation dan Value) dinyatakan dengan histogram. Dengan demikian citra HSV dinyatakan dengan tiga buah histogram yang terdiri dari histogram H, S, dan V. Pembuatan histogram dilakukan pada citra RGB yang sudah di konversi ke model HSV.

2.6 Hue Saturation Value (HSV)

Model warna HSV mendefinisikan warna dalam terminologi Hue, Saturation, dan Value. Hue menyatakan warna sebenarnya, seperti merah, violet, dan kuning. Hue digunakan untuk membedakan warnawarna dan menentukan kemerahan (redness), kehijauan (greenness) dari cahaya. Nilai hue antara 0 sampai 1 berarti warna antara merah melewati kuning, hijau, cyan, biru dan magenta dan

kembali menjadi merah. Nilai saturation antara 0 sampai 1 berarti dari tidak tersaturutasi (keabuan) sampai tersaturisasi penuh (tidak putih), (Agaputra dan Wardani, 2013). Nilai value atau brightness antara 0 sampai 1 berarti warna semakin cerah seperti ditunjukkan pada gambar 2.2



Gambar 2.2 Ruang Warna HSV pada Matlab (Kadir dan Susanto, 2013)

Model HSV pertama kali diperkenalkan A. R. Smith pada tahun 1978, untuk mendapatkan nilai H, S, V berdasarkan R, G, dan B, terdapat beberapa cara (Kadir dan Susanto, 2013). Cara yang tersederhana menurut Acharya dan Ray adalah seperti berikut (Kadir dan Susanto, 2013).

$$H = \tan\left(\frac{3(G-B)}{(R-G)+(R-B)}\right)$$

$$S = 1 - \frac{\min(R,G,B)}{V}$$

$$V = \frac{R+G+B}{3}$$

Cara kedua menurut Acharya dan Ray (Kadir dan Susanto, 2013) terdapat pada rumus-rumus yang digunakan sebagai berikut:

$$r = \frac{R}{(R+G+B)}, g = \frac{G}{(R+G+B)}, b = \frac{B}{(R+G+B)}$$

$$V = \max(r, g, b)$$

$$S = \begin{cases} 0, & \text{jika } V = 0 \\ 1 - \frac{\min(r,g,b)}{V}, & V > 0 \end{cases}$$

$$H = \begin{cases} 0, & \text{jika } S = 0 \\ \frac{60 \cdot (g-b)}{S+V}, & \text{jika } V = r \\ 60 \cdot \left[2 + \frac{b-r}{S+V}\right], & \text{jika } V = g \\ 60 \cdot \left[4 + \frac{r-g}{S+V}\right], & \text{jika } V = b \end{cases}$$

$$H = H + 360 \text{ jika } H < 0$$

Metode HSV (Hue Saturation Value) HSV memiliki karakteristik pokok dari warna antara lain :

- a. Hue digunakan untuk menentukan kemerahan (redness), kehijauan (greenness) dan sebagainya.
- b. Saturation adalah kemurnian atau kekuatan warna.
- c. Value kecerahan dari warna nilainya berkisar dari 0-100%. Apabila nilainya 0 maka warnanya adalah hitam, semakin besar nilai maka akan semakin cerah dan muncul variasi-variasi baru dari warna tersebut.

Menurut Zhou (Kadir dan susanto, 2013), manusia sebenarnya melihat warna adalah karena cahaya yang dipantulkan oleh objek. Dalam hal ini, spektrum cahaya kromatis berkisar antara 400-700 nm. Istilah kromatis berarti kualitas warna cahaya yang ditentukan oleh panjang gelombang. Karakteristik persepsi mata manusia dalam yang membedakan antara satu warna dengan warna yang lain berupa hue, saturation, dan brightness (Kadir dan Susanto, 2013).

Menurut Crane (Kadir dan Santosa, 2013), hue merujuk ke warna yang dikenal manusia, seperti merah dan hijau. Properti ini mencerminkan warna yang ditangkap oleh mata manusia yang menanggapi berbagai nilai panjang gelombang cahaya. Sebagai contoh, bila mata menangkap panjang gelombang antara 430 dan 480 nanometer, sensasi yang diterima adalah warna biru, sedangkan jika panjang gelombang berkisar antara 570 sampai dengan 600 nm, warna yang terlihat adalah kuning, sedang campuran merah dan hijau terlihat kuning. Saturation menyatakan tingkat kemurnian warna atau seberapa banyak cahaya putih yang tercampur dengan hue. Setiap warna murni bersaturasi 100% dan tidak mengandung cahaya putih sama sekali. Dengan kata lain, suatu warna murni yang bercampur dengan cahaya putih memiliki saturasi antara 0 dan 100%. Brightness atau kadang disebut lightness (kecerahan) menyatakan intensitas pantulan objek yang diterima mata. Intensitas dapat dinyatakan sebagai perubahan warna putih menuju abu-abu dan terakhir mencapai ke warna hitam, atau yang dikenal dengan istilah aras keabuan. Perlu diketahui, istilah kromatik berarti gabungan antara hue dan saturation dan istilah akromatik merujuk ke kecerahan.

Berdasarkan Gonzalez dan Woods (Kadir dan Susanto, 2013) mendefinisikan ruang warna (atau kadang disebut sistem warna atau model warna) sebagai suatu spesifikasi sistem koordinat dan suatu sub ruang dalam sistem tersebut dengan setiap warna dinyatakan dengan satu titik di dalamnya. Tujuan dibentuknya ruang warna adalah untuk memfasilitasi spesifikasi warna dalam bentuk suatu standar. Ruang warna yang paling dikenal pada perangkat komputer adalah RGB, yang sesuai dengan watak manusia dalam menangkap warna. Namun, kemudian dibuat banyak ruang warna, antara lain HSI, CMY, LUV, HSV, dan YIQ. HSV (Hue Saturation Value) terkadang dinamakan HSB (Hue Saturation Brightness).

2.7 K-Nearest Neighbor (K-NN)

K-Nearest Neighbor (K-NN) merupakan algoritma berjenis supervised yaitu algoritma yang hasil dari query instance baru diklasifikasikan berdasarkan mayoritas dari kategori pada K-NN. Atau garis besarnya algoritma ini mengklasifikasi objek berdasarkan atribut atau training sample. Algoritma ini tidak menggunakan pencocokan antar objek tetapi berdasarkan memori yang telah dimasukan sebelumnya. Algoritma ini menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari query instance yang baru. K-NN termasuk algoritma yang sederhana karena bekerja hanya berdasarkan jarak terpendek dari query instance ke data training sample untuk menentukan K-NN-nya. (Hanselman D, 1998)

Algoritma K-Nearest Neighbor merupakan sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan jumlah data yang mirip (tetangga terdekat) dengan jumlah k yang telah ditentukan mengklasifikasikan ke dalam kelas baru (Furqan, 2020).

Klasifikasi dengan KNN adalah mengelompokkan data baru, dalam penelitian ini adalah data uji berdasarkan jarak ke beberapa data k tetangga terdekat terhadap data latih. Prinsip kerja KNN adalah menghitung jarak menggunakan jarak Euclidean. Jarak Euclidean digunakan untuk menghitung jarak antara dua vektor yang berfungsi menguji ukuran yang bisa digunakan sebagai interpretasi kedekatan jarak antara dua obyek yang direpresentasikan (McAndrew, 2004). Pada fase data

latih, algoritma ini hanya melakukan penyimpanan nilai-nilai fitur dan klasifikasi data latih. Pada fase data uji, ditentukan nilai k (nilai tetangga) terlebih dahulu. Setelah nilai k ditentukan, jarak dari nilai data uji terhadap seluruh nilai data latih dihitung, dan diambil sejumlah nilai k yang paling dekat, kemudian diurutkan berdasarkan data yang memiliki jarak terdekat. Selanjutnya pilih data sampai urutan ke k, dan tentukan kelas berdasarkan model dari data latih. Dengan rumus berikut :

$$d(x, y) = \sqrt{\sum_{i=1}^p (x_i - y_i)^2}$$

Keterangan :

$d(x,y)$: jarak antara data uji dengan data latih

x_i : fitur ke i dari data uji

y_i : fitur ke i dari data latih

p : dimensi data

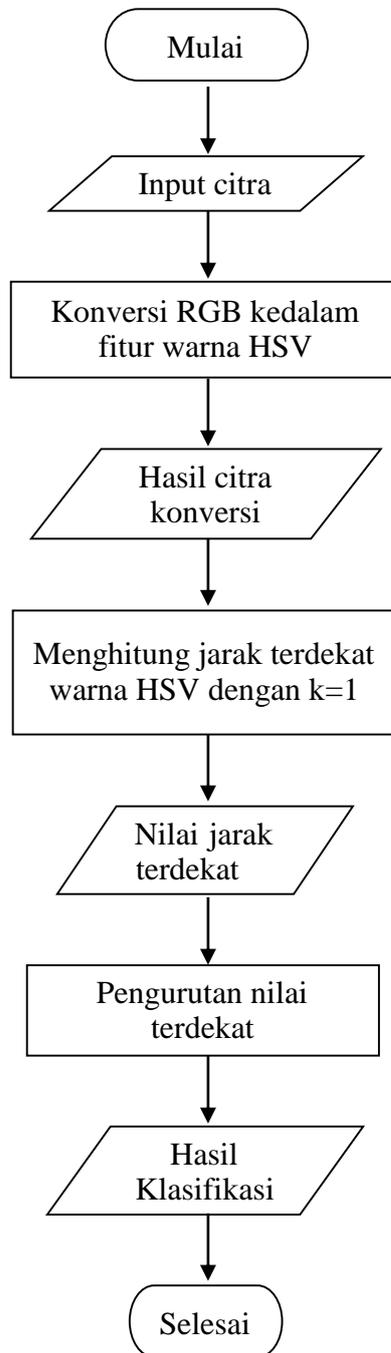
i : variable data

Kelebihan dari K-NN ini yaitu ketangguhan terhadap training data yang memiliki banyak noise dan efektif apabila training datanya besar, lebih sederhana dan mudah dipelajari. Sedangkan kelemahan dari K-NN ialah perlu menentukan nilai dari parameter k (jumlah dari tetangga terdekat), biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap sampel uji pada keseluruhan sampel latih.

Klasifikasi KNN dilakukan dengan mencari k buah tetangga terdekat dari data uji dan memilih kelas dengan anggota terbanyak. Adapun langkah-langkah klasifikasi KNN adalah sebagai berikut :

1. Menentukan parameter k (jumlah tetangga paling dekat)
2. Menghitung kuadrat jarak Euclidean (*query instance*) masing-masing objek terhadap data sampel yang diberikan
3. Kemudian mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak Euclid terkecil.
4. Mengumpulkan kategori y (klasifikasi *Nearest Neighbor* berdasarkan nilai k).

5. Dengan menggunakan kategori Nearest Neighbor yang paling mayoritas maka dapat diprediksi nilai query instance yang telah dihitung



Gambar 2.3 Flowchart Perhitungan K-NN

Gambar diatas merupakan alur dari metode k-nn dimana metode ini adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan ciri-ciri data pembelajaran (data latih) yang paling mendekati objek tersebut. Data latih dengan jarak terdekat dikatakan sebagai tetangga (Nearest Neighbor) kemudian diurutkan dari jarak terdekat sampai terjauh. Tiap tetangga dapat berbeda satu sama lain ataupun sejenis. Tetangga sejenis dengan jumlah terbanyak di antara K tetangga terdekat adalah data latih yang sesuai dengan objek yang diklasifikasikan.

2.8 Flowchart

Flowchart adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian (Anharku, 2009).

Pedoman-pedoman dalam membuat *flowchart* Jika seorang analis dan programmer akan membuat *flowchart*, ada beberapa petunjuk yang harus diperhatikan, seperti :

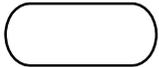
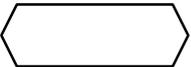
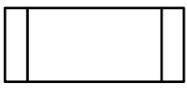
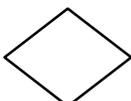
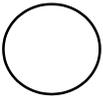
1. *Flowchart* digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja, misalkan Melakukan penggandaan diri.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan *range* dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada *flowchart* yang sama.

Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.

7. Gunakan simbol-simbol *flowchart* yang standar.

ANSI dan ISO merupakan penulis simbol *flowchart* yang sering di pakai dan bersifat standar. Berikut ini tabel 2.2 merupakan bagan alir beberapa simbol *flowchart* yang digunakan dalam menggambarkan suatu *flowchart*

Tabel 2.2 Simbol - Simbol Flowchart

Simbol	Nama	Fungsi
	<i>TERMINATOR</i>	Permulaan/ akhir program
	GARIS ALIR (<i>FLOW LINE</i>)	Arah aliran program
	<i>PREPARATION</i>	Proses inialisasi/ pemberian harga awal
	PROSES	Proses perhitungan/ proses pengolahan data
	<i>INPUT/OUTPUT DATA</i>	Proses input/ output data, parameter, informasi
	PREDEFINED PROCESS (SUB PROGRAM)	Permulaan sub program/ proses menjalankan subprogram
	DECISION	Perbandingan pernyataan, Penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	<i>ON PAGE CONNECTOR</i>	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	<i>OFF PAGE CONNECTOR</i>	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

2.9 Perhitungan Akurasi

Confusion matrix adalah tool yang digunakan untuk evaluasi model klasifikasi untuk memperkirakan objek yang benar atau salah. Sebuah matrix dari prediksi yang akan dibandingkan dengan kelas yang asli dari inputan atau dengan kata lain berisi informasi nilai aktual dan prediksi pada klasifikasi. Tingkat akurasi adalah tingkat keakuratan jaringan yang telah dibuat dalam mengenali inputan citra yang diberikan sehingga menghasilkan outputan yang benar. Adapun perhitungan tingkat akurasi pada confusion matriks dihitung dengan rumus sebagai berikut:

$$Accuracy = \frac{\text{jumlah data benar}}{\text{jumlah data keseluruhan}} \times 100\%$$

2.10 Penelitian Terkait

Berikut ini adalah tabel tentang penelitian penelitian terkait, yaitu sebagai berikut:

Tabel 2.3 Penelitian Terkait

No	Judul dan Tahun	Penulis	Ringkasan
1	Sistem klasifikasi jenis pisang berdasarkan ciri warna hsv menggunakan metode KNN. 2019	Zeni dwi lestari, dkk	Penelitian ini menggunakan metode Hue Saturation Value (HSV) untuk ekstraksi warna dan K-Nearest Neighbor sebagai klasifikasi. Penelitian yang dilakukan Zeni dkk terdiri dari 100 buah pisang diantaranya 50 pisang untuk data testing dan 50 pisang untuk data training, kemudian dibagi menjadi 5 kelas diantaranya yaitu kelas pisang ijo sebanyak 15 pisang, kelas sobo pipit sebanyak 10 pisang, kelas tandes sebanyak 5 pisang, kelas raja uli sebanyak 10 pisang dan kelas raja sebanyak 10 pisang. Dari pengujian sistem yang dilakukan menghasilkan akurasi yang cukup baik yaitu sebesar 82%
2	Pengukuran tingkat kematangan kopi robusta menggunakan algoritma KNN. 2018	Eko rachmawanto dan Abu salam	Penelitian ini menggunakan metode Hue Saturation Value (HSV) untuk proses ekstraksi ciri warna, kemudian menggunakan metode klasifikasi K-Nearest Neighbor. Data yang

			digunakan sebanyak 105 citra buah kopi robusta dengan membagi sebanyak 75 citra untuk data latih dan 30 citra untuk data uji. Penelitian ini berhasil dilakukan dengan mendapatkan akurasi yang tinggi sebesar 93%
3	Sistem klasifikasi jenis tanaman hias daun Philodendion Menggunakan Metode K-Nearest Neighbor Berdasarkan Nilai Hue, Saturation, dan Value (HSV) 2016	Deni syahid, dkk	Pada penelitian yang dilakukan Deni syahid, dkk menggunakan sampel citra data latih sebanyak 5 citra dan data uji sebanyak 10 citra. Akurasi sistem yang didapatkan mencapai 92%

Berdasarkan dari paparan penelitian diatas, didapatkan bahwa metode HSV merupakan metode yang telah terbukti menjadi descriptor ekstraksi fitur warna yang efektif, dan metode KNN telah banyak digunakan sebagai metode klasifikasi dengan akurasi yang tinggi.

BAB III
METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

3.1.1 Tempat Penelitian

Tempat penelitian yang dilakukan peneliti adalah di Laboratorium Multimedia Fakultas Sains dan Teknologi, Jl. IAIN No. 1 Medan, Sumatera Utara

3.1.2 Waktu Penelitian

Waktu penelitian yang dilakukan peneliti pada semester genap tahun ajaran 2019/2020, seperti tabel berikut ini:

Tabel 3.1 Waktu Penelitian

No	Kegiatan	Bulan/2020						
		Mei	Juni	Juli	Agt	Sept	Okt	Nov
1.	Tahap persiapan penelitian							
	a. Penyusunan dan pengajuan judul							
	b. Pengajuan proposal							
2.	Tahap pelaksanaan							
	a. Pengumpulan data							
	b. Analisis data							
3.	Tahap penyusunan laporan							

3.2 Bahan dan Alat Penelitian

Penelitian ini dilakukan dengan menggunakan sampe citra ikan, perangkat keras serta perangkat lunak sebagai berikut:

3.2.1 Perangkat Keras

Perangkat keras yang diperlukan peneliti untuk pembuatan sistem ini ialah sebagai berikut:

1. Labtop

Merk : *Asus*
 Penyimpanan : *disk drive 500GB*
 Memori : *Random Access Memory (RAM) 2GB*
 Processor : *Intel(R) Celeron(R) CPU N3060 @1.60GHz 1.60 GHz*

2. Smartphone

Merk : *Oppo*
 Kamera : *13 MP*
 Type : *A31*
 RAM : *4.00 GB*

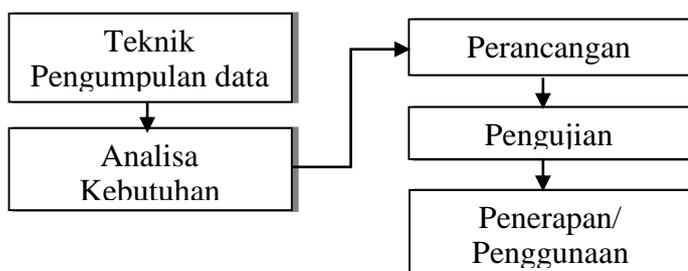
3.2.2 Perangkat Lunak

Perangkat lunak yang diperlukan peneliti untuk pembuatan sistem ini ialah sebagai berikut:

1. Operating System Windows 10 Pro 64 bit.
2. Matlab R2017

3.3 Prosedur Kerja

Tahap tahap prosedur kerja dalam pembuatan sistem ini ialah sebagai berikut:



Gambar 3.1. Tahap – Tahap Prosedur Kerja

3.3.1 Teknik Pengumpulan Data

Beberapa teknik pengumpulan data yang dilakukan peneliti dalam melakukan penelitian ini ialah sebagai berikut :

1. Studi Literatural

Teknik pengumpulan data yang peneliti lakukan pertama ialah studi literatul. Studi Literatural merupakan serangkaian kegiatan yang berkenaan metode pengumpulan daftar pustaka, membaca dan mencatat, serta mengolah bahan penelitian atau mencari referensi teori yang relevan dengan kasus atau permasalahan yang berhubungan dengan dengan penelitian ini.

2. Observasi

Observasi merupakan teknik pengumpulan data yang dilakukan peneliti dengan pengamatan langsung terhadap objek yang ada di lingkungan yang sedang berlangsung meliputi berbagai aktivitas perhatian terhadap kajian objek dengan menggunakan pengindraan.

3.3.2 Analisis Kebutuhan

Untuk membangun sebuah sistem tahap analisis sangat perlu dilakukan agar mendapatkan sebuah informasi kebutuhan apa saja yang digunakan demi mendukung sistem berjalan semestinya. Kebutuhan tersebut mencakup perangkat keras dan perangkat lunak.

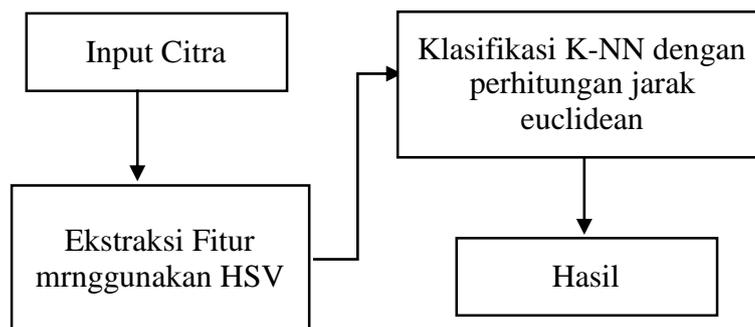
Ekstraksi fitur menggunakan Hue Saturation Value (HSV) dipilih karena memiliki kinerja paling baik untuk pengenalan warna, dan klasifikasi K-NN algoritma yang sederhana dimana algoritma ini hanya bekerja berdasarkan jarak terpendek dari query instance ke data training sample. Banyak jurnal yang membahas mengenai ekstraksi fitur/ciri, hal ini tentu saja membantu peneliti dalam membuat penelitian ini.

3.3.3 Perancangan

Perancangan adalah langkah pertama dalam fase pengembangan rekayasa produk atau sistem. Perancangan itu adalah proses penerapan berbagai teknik dan

prinsip yang bertujuan untuk mendefinisikan sebuah peralatan, satu proses atau satu sistem secara detail yang membolehkan dilakukan realisasi fisik. Fase ini adalah inti teknis dari proses rekayasa perangkat lunak.

Adapun diagram perencanaan sistem adalah sebagai berikut



Gambar 3.2 Diagram Perencanaan Sistem

Diagram atas merupakan alur dari sistem untuk melakukan sebuah ekstraksi fitur untuk mengekstraksi fitur warna pada citra ikan dengan menggunakan metode *hue saturation value*. Dimulai dari menginput citra yang didapatkan dari akuisisi citra, yaitu pengambilan image menggunakan kamera smartphone, lalu mengkonversi citra *RGB* menjadi citra warna *HSV* kemudian dilakukan ekstraksi fitur warna menggunakan metode *Hue Saturation Value (HSV)*, dan setelah dilakukan ekstraksi fitur warna kemudian di klasifikasi menggunakan metode *k-nn*, dengan perhitungan menggunakan jarak *euclidean distance*. Maka hasil dari ekstraksi fitur warna *HSV* dan klasifikasi *KNN* diketahui dan proses klasifikasi kesegaran ikan selesai.

3.3.4 Pengujian

Pengujian sistem membahas mengenai hasil uji coba sistem yang telah di rancang. Pengujian bermaksud untuk mengetahui apakah sistem yang dibuat dapat berjalan dengan baik dan sudah memenuhi kriteria yang sesuai dengan tujuan perancangan sistem tersebut. Pengujian dilakukan oleh penguji dengan cara menggunakan sistem tersebut secara langsung.

3.3.5 Penerapan / Penggunaan

Penerapan/penggunaan sistem ini ialah menjelaskan sistem yang dibangun dengan menggunakan ekstraksi fitur warna pada bagian mata ikan dengan menerapkan metode *hsv* dan metode klasifikasi *k-nn* bermanfaat untuk membantu penguji dalam meminimalisir kesalahan dalam mengetahui tingkat kesegaran ikan tersebut khususnya kepada konsumen.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Pembahasan

Ada beberapa tahapan yang akan dilakukan dalam penelitian ini untuk pemecahan masalah berdasarkan rumusan masalah, diantaranya ialah tahap analisis data, representasi data, hasil analisis data, perancangan sistem dan pengujian sistem sebagai berikut.

4.1.1. Analisis Data

Analisis data untuk klasifikasi kesegaran ikan berdasarkan ekstraksi fitur menggunakan metode *K-NN* dan *HSV* dilakukan pengumpulan data set citra ikan yang akan diolah menggunakan kedua metode tersebut. Analisis kebutuhan terdiri dari kebutuhan proses, kebutuhan masukan dan kebutuhan keluaran. Analisis kebutuhan proses, yaitu menjelaskan bagaimana sistem akan bekerja, proses-proses apa yang digunakan, mulai dari masuknya data input yang kemudian diproses oleh sistem hingga menjadi data output (tampilan akhir sistem).

Pada penelitian ini dilakukan analisis data yang bertujuan untuk mengklasifikasi hasil ekstraksi fitur warna RGB dan di konversi ke model warna HSV dan kemudian menggunakan *k*-nearest neighbor berdasarkan jarak tetangga yang paling terdekat.

Implementasi matlab menggunakan metode *knn* dan model warna *hsv* untuk menganalisis kesegaran ikan menggunakan matlab 2017. Ada beberapa langkah program yang dilakukan yaitu :

- a. Menginput citra ikan.
- b. Citra ikan diekstraksi ke fitur warna menggunakan metode *HSV*
- c. Identifikasi/Klasifikasi kesegaran ikan menggunakan metode *K-NN* melalui perhitungan jarak *euclidean*.
- d. Hasil klasifikasi citra menggunakan metode *HSV* dan *KNN*

4.1.2. Representasi Data

Data set yang dibutuhkan untuk perhitungan manual sebanyak 4 sampel citra yang terdiri 2 sampel citra untuk klasifikasi ikan segar dan 2 sampel untuk klasifikasi ikan tidak segar. Berikut adalah gambar dari citra sampel ikan untuk proses hitungan manual :



Gambar 4.1 Citra Ikan Sampel

Dari gambar di atas, diketahui citra ikan segar dan citra ikan tidak segar sebanyak 2 citra masing-masing klasifikasi. Kemudian melakukan ekstraksi citra RGB menjadi citra HSV. Pada tahap klasifikasi kesegaran ikan dilakukan dengan proses pelatihan dan pengujian. Proses pelatihan adalah proses dimana keseluruhan citra ikan diekstraksi dengan HSV dan disimpan ke dalam dataset. Sedangkan proses pengujian adalah melakukan pengujian klasifikasi kesegaran ikan dengan ekstraksi HSV dan melakukan pendekatan nilai dengan dataset latih untuk mendapatkan klasifikasi ikan.

1. Proses Pelatihan HSV

Berdasarkan dari gambar 4.1, untuk memudahkan perhitungan manual ekstraksi citra RGB menjadi HSV maka diambil sampel citra A, B, dan C dengan resolusi 3x3 pixel. Adapun nilai RGB dari citra ikan segar 1 yang diklasifikasikan menjadi menjadi citra ikan segar adalah sebagai berikut :

Tabel 4.1 Nilai RGB 3x3 Citra Ikan Segar 1

X/y	0	1	2
0	20	132	82
	156	5	18
	123	82	10
1	58	14	105
	53	121	8
	38	71	76
2	30	66	21
	130	37	15
	92	110	40

Kemudian mencari nilai rata-rata Red, Green dan Blue dari citra 3x3 ikan segar 1 sebagai berikut:

Nilai Rata-Rata RGB Citra Ikan Segar 1

$$R = \frac{20+132+82+58+14+105+30+66+21}{9} = 58,6$$

$$G = \frac{156+5+18+53+121+8+130+37+15}{9} = 60,3$$

$$B = \frac{123+82+10+38+71+76+92+110+40}{9} = 71,3$$

Nilai RGB dari citra ikan segar 2 yang akan diklasifikasikan menjadi citra ikan segar adalah sebagai berikut :

Tabel 4.2 Nilai RGB 3x3 Citra Ikan Segar 2

X/y	0	1	2
0	25	130	77
	149	10	12
	119	78	8
1	61	19	100
	48	119	15
	33	67	72
2	35	59	19
	128	32	13
	87	107	37

Selanjutnya adalah mencari nilai rata-rata Red, Green dan Blue pada citra 3x3 ikan segar 2 sebagai berikut:

Nilai Rata-Rata RGB Citra Ikan Segar 2

$$R = \frac{25+130+77+61+19+100+35+59+19}{9} = 58,3$$

$$G = \frac{149+10+12+48+119+15+128+32+13}{9} = 58,4$$

$$B = \frac{119+78+8+33+67+72+87+107+37}{9} = 67,5$$

Adapun nilai RGB dari citra ikan tidak segar 1 yang diklasifikasikan menjadi citra ikan tidak segar ialah sebagai berikut :

Tabel 4.3 Nilai RGB 3x3 Citra Ikan Tidak Segar 1

X/y	0	1	2
0	87	10	45
	87	121	105
	43	54	5
1	131	114	23
	28	8	41
	177	78	69
2	111	4	80
	24	72	93
	33	78	143

Kemudian mencari nilai rata-rata Red, Green dan Blue pada citra 3x3 ikan tidak segar 1 sebagai berikut:

Nilai Rata-Rata RGB Citra Ikan Tidak Segar 1

$$R = \frac{87+10+45+131+114+23+111+4+80}{9} = 67,2$$

$$G = \frac{87+121+105+28+8+41+24+72+93}{9} = 64,3$$

$$B = \frac{43+54+5+177+78+69+33+78+143}{9} = 75,5$$

Adapun nilai RGB dari citra ikan tidak segar 2 yang akan diklasifikasikan menjadi citra ikan tidak segar ialah sebagai berikut :

Tabel 4.4 Nilai RGB 3x3 Citra Ikan Tidak Segar 2

X/y	0	1	2
0	91	5	40
	81	119	101
	39	48	7
1	128	120	19
	30	10	43
	169	82	71
2	107	12	75
	20	85	88
	37	64	139

Kemudian mencari nilai rata-rata Red, Green dan Blue pada citra 3x3 ikan tidak segar 2 sebagai berikut:

Nilai Rata-Rata RGB Citra Ikan Tidak Segar 2

$$R = \frac{91+5+40+128+120+19+107+12+75}{9} = 64,1$$

$$G = \frac{81+119+101+30+10+43+20+85+88}{9} = 72,8$$

$$B = \frac{39+48+7+169+82+71+37+64+139}{9} = 51,2$$

Berdasarkan pencarian nilai rata-rata citra RGB 3x3 setiap citra ikan maka didapatkan hasil rata-rata nilai RGB citra ikan segar dan tidak segar seperti berikut ini:

Tabel 4.5 Nilai Rata-Rata RGB Citra Ikan

Nama Citra	Warna	Nilai Rata-rata
Citra Ikan Segar 1	RED	58,6
	GREEN	60,3
	BLUE	71,3
Citra Ikan Segar 2	RED	58,3
	GREEN	58,4
	BLUE	67,5
Citra Ikan tidak Segar 1	RED	67,2
	GREEN	64,3
	BLUE	75,5
Citra Ikan tidak Segar 2	RED	64,1
	GREEN	72,8
	BLUE	51,2

Berdasarkan pada tabel di atas, nilai RGB setiap citra telah didapatkan, proses selanjutnya melakukan konversi nilai RGB citra ikan segar dan tidak segar kedalam bentuk nilai HSV. Rumus HSV yang digunakan adalah sebagai berikut :

$$H = \tan\left(\frac{3(G-B)}{(R-G)+(R-B)}\right)$$

$$S = 1 - \frac{\min(R,G,B)}{V}$$

$$V = \frac{R+G+B}{3}$$

Berdasarkan dari rumus diatas dijelaskan bahwa tahap pertama adalah mencari nilai H, kemudian mencari nilai V untuk mendapatkan nilai S. Adapun langkah-langkah konversi citra RGB menjadi citra HSV dengan citra ikan segar dan tidak segar sampel sebesar 3x3 pixel dapat dilihat sebagai berikut:

- a. Proses Konversi Citra Ikan Segar 1

Diketahui nilai RGB citra Ikan segar 1 adalah :

$$R = 58,6$$

$$G = 60,3$$

$$B = 71,3$$

Kemudian proses konversi kedalam HSV sebagai berikut:

$$H = \tan \frac{3 \times (60,3 - 71,3)}{(58,6 - 60,3) + (58,6 - 71,3)} = 0,04$$

$$V = \frac{58,6 + 60,3 + 71,3}{3} = 63,444$$

$$S = 1 - \frac{58,6}{63,4} = 0,075$$

Dari proses di atas, didapati nilai HSV citra RGB ikan segar 1.

b. Proses Konversi HSV Citra Ikan Segar 2

Diketahui nilai RGB dari citra ikan segar 2 adalah berikut:

$$R = 58,3$$

$$G = 58,4$$

$$B = 67,5$$

Kemudian proses konversi kedalam HSV sebagai berikut:

$$H = \tan \frac{3 \times (58,3 - 67,5)}{(58,3 - 58,4) + (58,3 - 67,5)} = 0,051$$

$$V = \frac{58,3 + 58,4 + 67,5}{3} = 61,444$$

$$S = 1 - \frac{58,3}{61,4} = 0,048$$

Dari proses diatas, didapati nilai HSV citra RGB ikan segar 2

c. Proses Konversi HSV Citra Ikan Tidak Segar 1

Diketahui nilai RGB citra ikan tidak segar 1 adalah berikut:

$$R = 67,2$$

$$G = 64,3$$

$$B = 75,5$$

Kemudian proses konversi kedalam HSV sebagai berikut:

$$H = \tan \frac{3 \times (64,3 - 75,5)}{(67,2 - 64,3) + (67,2 - 75,5)} = 0,108$$

$$V = \frac{67,2 + 64,3 + 75,5}{3} = 69,037$$

$$S = 1 - \frac{64,3}{69,03} = 0,068$$

Dari proses diatas, didapati nilai HSV citra RGB ikan tidak segar 1.

d. Proses Konversi HSV Citra Ikan Tidak Segar 2

Diketahui nilai RGB citra ikan tidak segar 2 adalah :

$$R = 64,1$$

$$G = 72,8$$

$$B = 51,2$$

Kemudian proses konversi kedalam HSV sebagai berikut:

$$H = \tan \frac{3 \times (51,2 - 72,8)}{(64,1 - 72,8) + (64,1 - 51,2)} = 0,283$$

$$V = \frac{64,1 + 72,8 + 51,2}{3} = 62,74$$

$$S = 1 - \frac{72,8}{62,74} = -0,161$$

Dari proses diatas, didapati nilai HSV citra RGB ikan tidak segar 2

Proses pelatihan telah dilakukan secara manual dengan citra sampel ikan segar dan tidak segar dengan ukuran 3x3 pixel. Kemudian setiap nilai HSV citra disimpan kedalam dataset, sehingga dataset citra HSV ikan segar dan tidak segar dapat dilihat pada tabel di bawah ini:

Tabel 4.6 Nilai HSV Citra Ikan

Nama Citra	Klasifikasi	Ciri	Nilai
Citra Ikan Segar 1	Ikan Segar	H	0,04
		S	0,075
		V	63,44
Citra Ikan Segar 2	Ikan Segar	H	0,051
		S	0,048
		V	61,44

Tabel Lanjutan 4.6 Nilai HSV Citra Ikan

Nama Citra	Klasifikasi	Ciri	Nilai
Citra Ikan tidak Segar 1	Ikan tidak Segar	H	0,108
		S	0,068
		V	69,03
Citra Ikan tidak Segar 2	Ikan tidak Segar	H	0,283
		S	-0,161
		V	62,74

4.1.3. Hasil Analisis Data

Setelah didapatkannya dataset citra HSV pada tabel 4.6, proses selanjutnya melakukan pengujian klasifikasi kesegaran ikan. Adapun pada proses hitungan manual klasifikasi kesegaran ikan dilakukan dengan citra ikan uji dengan klasifikasi “ikan segar”. Berikut ini merupakan citra ikan untuk sampel uji yang akan diklasifikasi:

**Gambar 4.2** Citra Ikan Uji

Berdasarkan pada gambar citra di atas, untuk memudahkan perhitungan manual pada saat proses klasifikasi kesegaran ikan maka diambilah sampel citra uji dari gambar 4.2 dengan resolusi 3x3 pixel. Adapun nilai RGB dari citra uji adalah sebagai berikut :

Tabel 4.7 Nilai RGB 3x3 Citra Uji

X/y	0	1	2
0	31	132	68
	143	15	7
	123	74	12
1	65	23	118
	36	111	20
	28	66	80
2	31	53	21
	126	29	10
	88	115	39

Kemudian mencari nilai rata-rata Red, Green dan Blue pada citra uji sebagai berikut:

$$R = \frac{31+132+68+65+23+118+31+53+21}{9} = 60,2$$

$$G = \frac{143+15+7+36+111+20+126+29+10}{9} = 55,2$$

$$B = \frac{123+74+12+28+66+80+88+115+39}{9} = 69,4$$

Proses selanjutnya ialah mencari nilai HSV berdasarkan nilai RGB pada citra uji sebagai berikut:

$$H = \tan \frac{3 \times (55,2 - 69,4)}{(60,2 - 55,2) + (60,2 - 69,4)} = 0,178$$

$$V = \frac{60,2 + 55,2 + 69,4}{3} = 61,629$$

$$S = 1 - \frac{55,2}{61,6} = 0,103$$

Setelah proses mencari nilai HSV selesai dengan nilai H= 0,178, S=0,103 dan nilai V=61,629 untuk citra uji, kemudian adalah melakukan klasifikasi kesegaran ikan dengan membandingkan jarak terdekat dari dataset nilai HSV citra latih ikan segar dan tidak segar yang terdapat di tabel 4.6, proses perhitungannya dengan rumus

$$d(x, y) = \sqrt{\sum_{i=1}^p (x_i - y_i)^2}$$

$$\text{Segar 1} \sqrt{(0,178 - 0,04)^2 + (61,629 - 63,44)^2 + (0,103 - 0,075)^2} = 1,82$$

$$\text{Segar 2} \sqrt{(0,178 - 0,051)^2 + (61,629 - 61,44)^2 + (0,103 - 0,048)^2} = 0,231$$

$$\text{Tidak Segar 1} \sqrt{(0,178 - 0,108)^2 + (61,629 - 69,037)^2 + (0,103 - 0,068)^2} = 7,407$$

$$\text{Tidak Segar 2 } \sqrt{(0,178 - 0,283)^2 + (61,629 - 62,74)^2 + (0,103 - (-0,161))^2} = 1,147$$

Berdasarkan perhitungan jarak *euclidean distance*, kemudian dirangkum dengan jarak paling dekat yaitu dengan nilai $K=1$, maka menghasilkan klasifikasi kesegaran ikan seperti pada tabel di bawah ini:

Tabel 4.8 Data Jarak Terdekat Berdasarkan Citra Ikan Uji

Data	Jarak Dengan Data Baru	Tingkat Kesegaran
Citra Ikan Segar 1	1,82	Ikan Segar
Citra Ikan Segar 2	0,231	Ikan Segar
Citra Ikan Tidak Segar 1	7,407	Ikan Tidak Segar
Citra Ikan Tidak Segar 2	1,147	Ikan Tidak Segar

Berdasarkan Tabel 4.8 diatas hasil klasifikasi kesegaran ikan dengan perhitungan jarak tetangga yang paling terdekat yaitu citra ikan segar 2 dengan klasifikasi “Ikan Segar”, sehingga data citra ikan uji yang diproses masuk kedalam klasifikasi sebuah citra “Ikan Segar”.

4.1.4. Perancangan Sistem

Pada penelitian ini dilakukan beberapa perancangan sistem ataupun gambaran dari sebuah sistem yang akan dibangun diantaranya ialah sebagai berikut:.

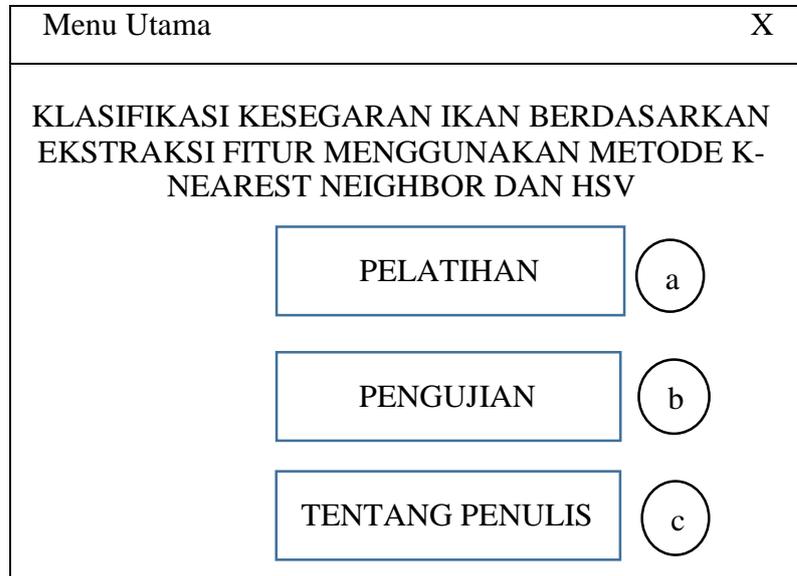
1. Perancangan Antar Muka

Sebelum diimplementasikan ke dalam sebuah program aplikasi, hal yang harus di lakukan yaitu merancang sistem klasifikasi kesegaran ikan agar sistem tersebut dapat berjalan sebagaimana mestinya. Beberapa perancangan sistem yang akan dibuat terdiri dari rancangan menu utama, rancangan menu pelatihan, rancangan menu pengujian dan rancangan menu tentang penulis. Berikut ini merupakan keseluruhan rancangan *interface* sistem yang akan dibangun:

a. Rancangan Menu Utama

Menu utama merupakan sebuah menu yang akan tampil ketika pertama kali program dijalankan. Menu utama juga sebagai navigasi untuk ke menu-menu

yang lain. Berikut ini adalah rancangan menu utama dapat dilihat pada gambar di bawah:

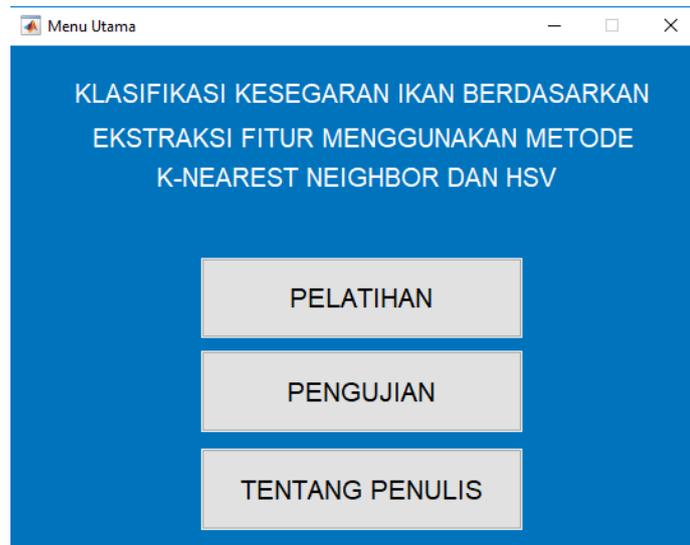


Gambar 4.3 Rancangan Menu Utama

Pada gambar diatas, rancangan menu utama terdiri dari button pelatihan, pengujian, dan button tentang penulis, berikut keterangan dari ketiga button tersebut:

- a. Button yang berfungsi untuk menampilkan menu pelatihan klasifikasi kesegaran ikan
- b. *Button* yang berfungsi untuk menampilkan menu pengujian klasifikasi kesegaran ikan
- c. *Button* yang berfungsi untuk menampilkan menu informasi tentang penulis

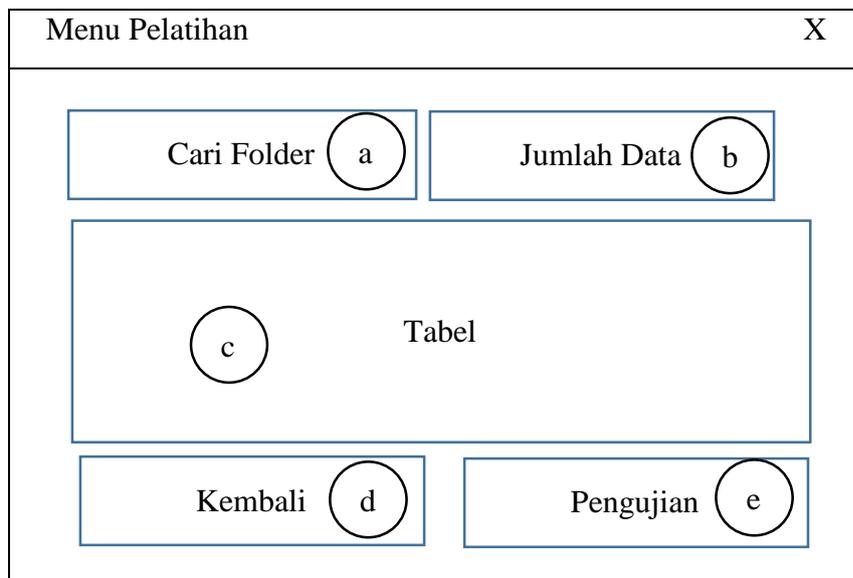
Berikut ini merupakan tampilan program menu utama yang telah dibangun kedalam aplikasi menggunakan matlab:



Gambar 4.4 Tampilan Menu Utama

b. Rancangan Menu Pelatihan

Menu pelatihan merupakan sebuah menu yang akan tampil ketika pengguna akan melakukan pelatihan untuk menyimpan dataset citra ikan yang telah diekstraksi. Berikut ini adalah rancangan menu pelatihan dapat dilihat pada gambar dibawah:

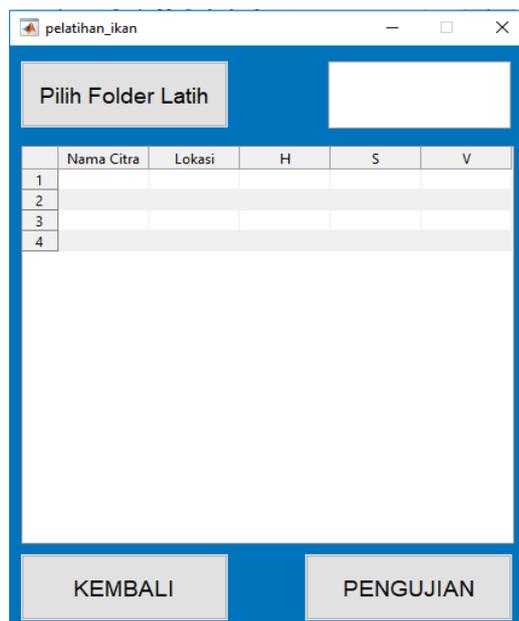


Gambar 4.5 Rancangan Menu Pelatihan

Pada gambar diatas, rancangan menu pelatihan terdiri dari *button* cari folder, *edittext* jumlah data, *uitabel* tabel, *button* kembali dan *button* pengujian. Berikut keterangan dari gambar rancangan menu pelatihan.

- a. *Button* ini berfungsi untuk menampilkan pop up pencarian folder tempat citra ikan disimpan
- b. *Edittext* berfungsi untuk menampilkan jumlah data citra yang berada dalam folder yang akan dipilih
- c. *Uitabel* disini berfungsi menampilkan informasi citra dengan nilai HSV pada setiap citra yang berada pada folder yang akan dipilih
- d. *Button* ini berfungsi untuk kembali ke menu utama
- e. *Button* ini berfungsi menampilkan menu pengujian

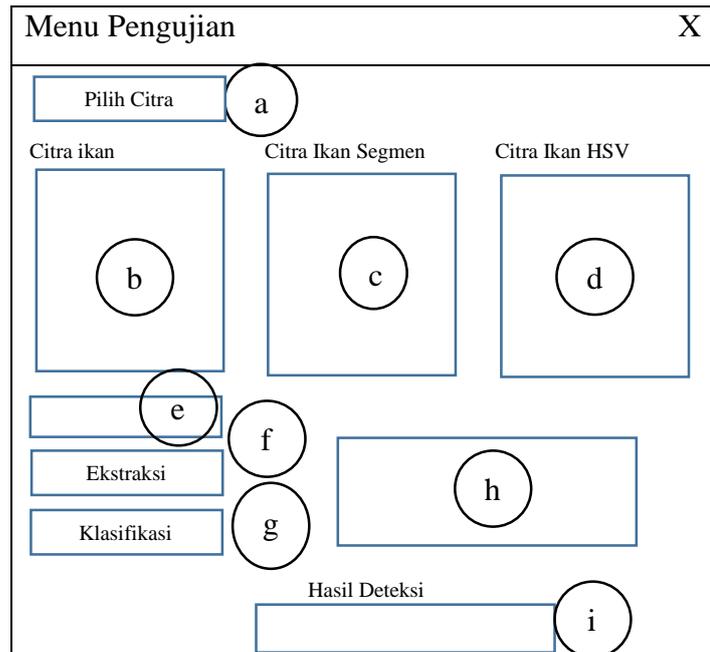
Berikut ini merupakan tampilan program menu pelatihan yang telah dibangun kedalam aplikasi menggunakan matlab dapat dilihat sebagai berikut:



Gambar 4.6 Tampilan Menu Pelatihan

c. Rancangan Menu Pengujian

Menu pengujian merupakan sebuah menu yang akan tampil ketika pengguna akan melakukan pengujian klasifikasi kesegaran ikan. Berikut ini adalah rancangan pengujian dapat dilihat pada gambar dibawah:



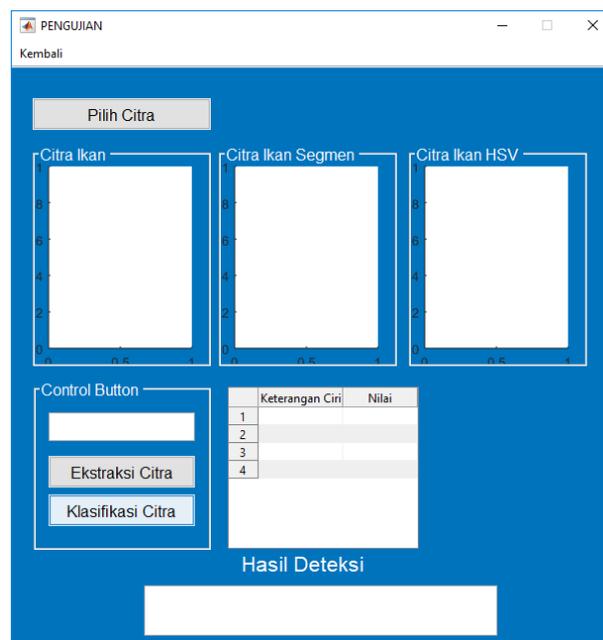
Gambar 4.7 Rancangan Menu Pengujian

Pada gambar diatas rancangan menu pengujian terdiri dari *button* pilih citra, *axes* citra ikan, *axes* citra ikan segmen, *axes* citra ikan *hsv*, *edittext*, *button* ekstraksi, *button* klasifikasi, dan *uitabel* hasil deteksi. Berikut keterangan dari gambar rancangan menu pengujian.

- Button* ini berfungsi untuk memilih dan mencari citra ikan yang akan diuji
- Axes* pertama berfungsi menampung data citra ikan yang telah dipilih
- Axes* kedua berfungsi menampung data citra ikan segmen
- Axes* ketiga berfungsi menampung data citra ikan ekstraksi *HSV*
- Edittext* ini berfungsi menampilkan informasi nama dari citra ikan yang dipilih
- Button* ini berfungsi untuk melakukan ekstraksi *HSV* citra ikan
- Button* ini berfungsi melakukan klasifikasi citra kesegaran ikan yang dipilih
- Uitabel* ini berfungsi untuk menampilkan nilai ekstraksi citra ikan

- i. *Edittext* berfungsi untuk menampilkan informasi hasil klasifikasi citra kesegaran ikan yang dipilih

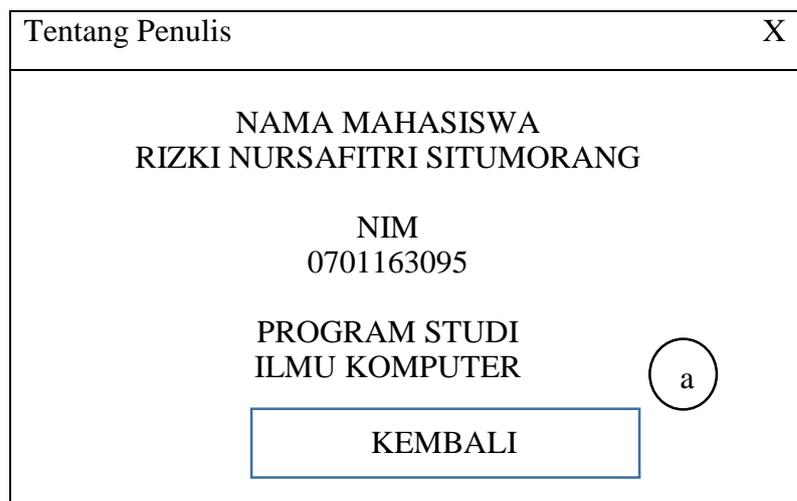
Berikut ini merupakan tampilan program menu pengujian yang telah dibangun kedalam aplikasi menggunakan matlab dapat dilihat sebagai berikut:



Gambar 4.8 Tampilan Menu Pengujian

d. Rancangan Tentang Penulis

Menu info merupakan sebuah menu yang menampilkan ketika pengguna melihat info tentang penulis. Berikut ini rancangan menu tentang penulis dapat dilihat pada gambar dibawah:

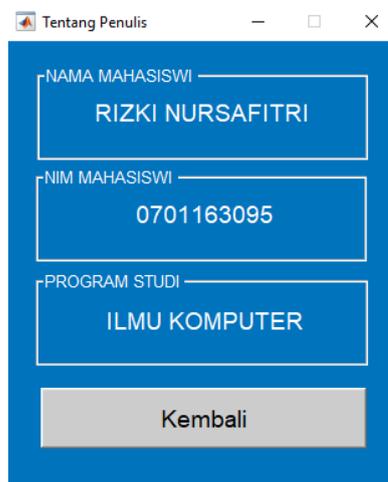


Gambar 4.9 Rancangan Menu Tentang Penulis

Berikut keterangan dari gambar rancangan menu tentang penulis:

- a. *Button* ini berfungsi untuk kembali pada menu utama

Berikut ini merupakan tampilan program menu tentang penulis yang telah dibangun kedalam sebuah aplikasi menggunakan matlab dapat dilihat sebagai berikut:



Gambar 4.10 Tampilan Menu Tentang Penulis

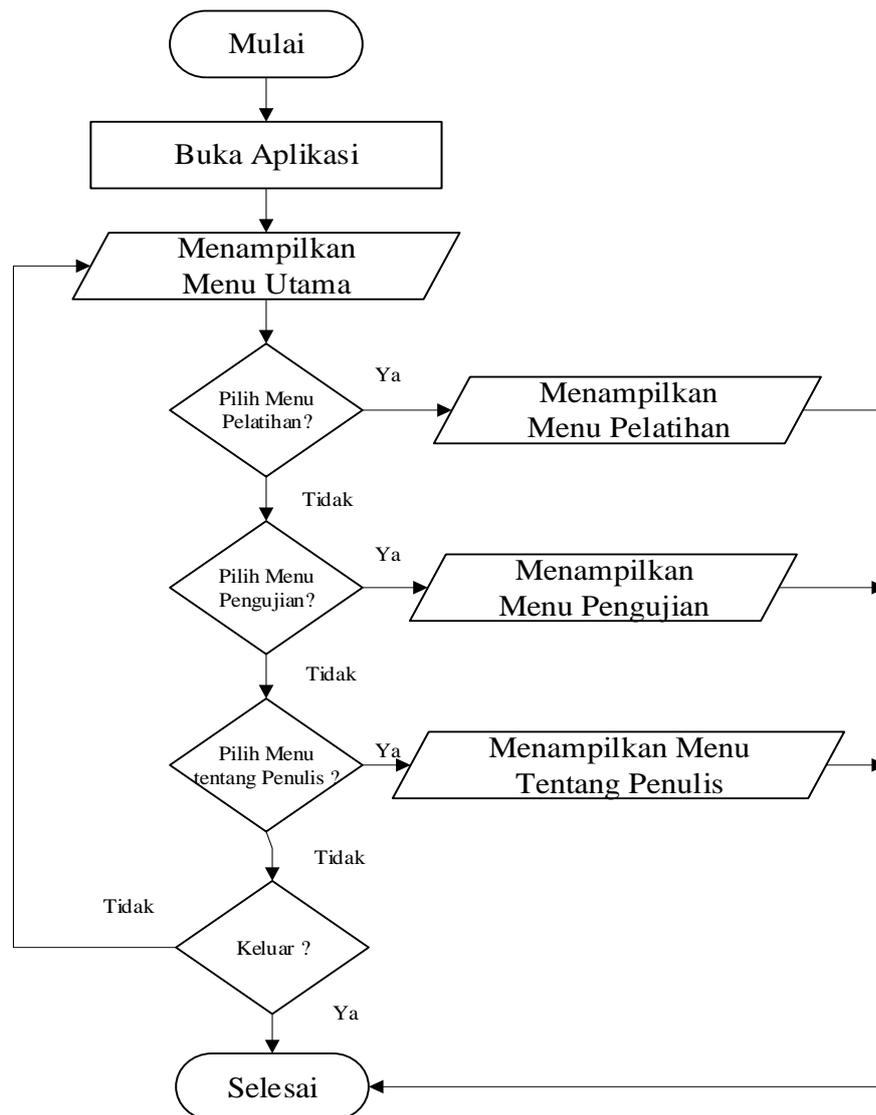
2. Flowchart Sistem

Flowchart sistem adalah *flowchart* yang berfungsi untuk menunjukkan alur proses dari sistem yang akan dibangun. Pada sistem ini *flowchart* yang dibutuhkan terdiri dari *flowchart* menu utama, *flowchart* pelatihan, *flowchart* pengujian dan

flowchart tentang penulis. Berikut ini keseluruhan dari *flowchart* sistem yang akan dibangun

1. *Flowchart* Menu Utama

Flowchart menu utama ini ialah alur proses ketika pengguna berada pada menu utama. Dibawah ini adalah rancangan dari *flowchart* menu utama:



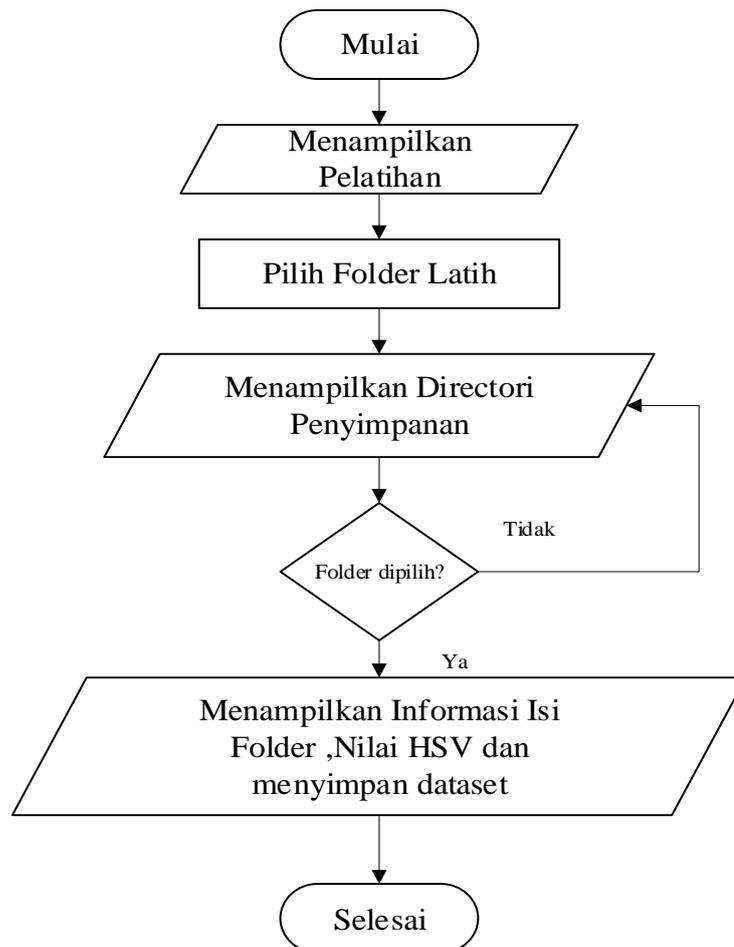
Gambar 4.11 *Flowchart* Menu Utama

Dari gambar *flowchart* diatas dijelaskan bahwa langkah pengguna yang akan membuka aplikasi ini dihadapkan dengan 3 menu diantaranya menu pelatihan digunakan untuk memproses pelatihan citra latih, menu pengujian digunakan untuk

mengklasifikasi kesegaran ikan, dan menu tentang penulis untuk menampilkan informasi tentang penulis penelitian ini.

2. Flowchart Pelatihan

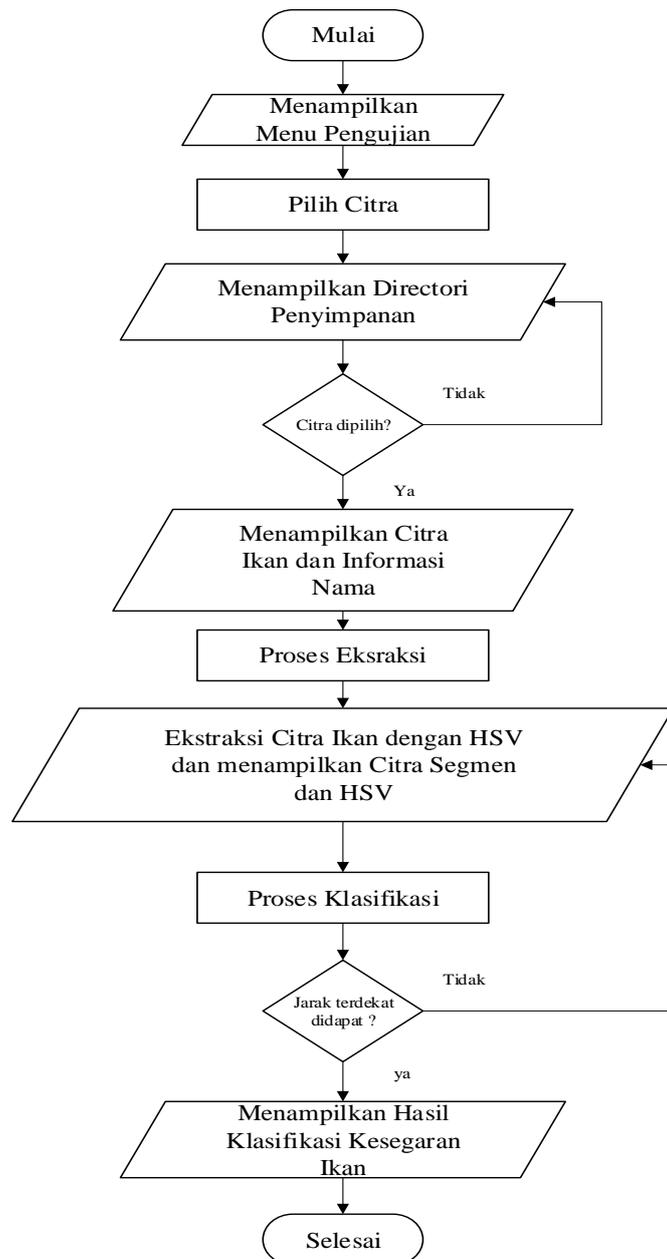
Flowchart pelatihan ini ialah alur proses ketika pengguna melakukan pelatihan terhadap citra ikan yang akan disimpan kedalam dataset. Dibawah ini adalah rancangan dari *flowchart* pelatihan:



Gambar 4.12 *Flowchart* Pelatihan

3. *Flowchart* Pengujian

Flowchart pengujian ini ialah alur proses ketika pengguna akan melakukan pengujian klasifikasi kesegaran ikan. Dibawah ini adalah rancangan dari *flowchart* pengujian:

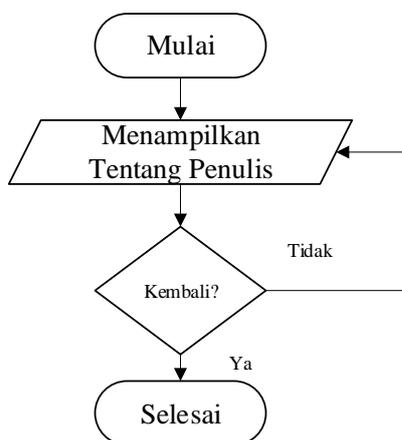


Gambar 4.13 *Flowchart* Pengujian

Dari gambar *flowchart* pengujian citra diatas dijelaskan bahwa langkah pertama pengguna melakukan pemilihan citra ikan yang akan diuji, kemudian diproses ekstraksi menggunakan metode HSV, hasil ekstraksi kemudian dicari nilai terdekat dengan dataset menggunakan KNN, sehingga menghasilkan klasifikasi kesegaran ikan yang sesuai dengan dataset.

4. *Flowchart* Tentang Penulis

Flowchart tentang penulis ini ialah alur proses ketika pengguna akan membuka menu tentang penulis. Dibawah ini adalah rancangan dari *flowchart* tentang penulis:



Gambar 4.14 *Flowchart* Tentang Penulis

Dari gambar *flowchart* tentang penulis diatas dijelaskan bahwa pengguna disajikan dengan informasi tentang penulis penelitian ini.

4.2. Hasil

Berikut ini akan dibahas mengenai hasil yang didapat pada penelitian ini yaitu pengujian dan penerapan sebagai berikut.

4.2.1. Pengujian

Setelah perancangan sistem selesai maka dilakukan pengujian untuk melihat sejauh mana sistem yang telah dibangun, apakah sesuai dengan yang diharapkan. Tahapan pada pengujian sistem ini dengan menyiapkan data latih dan data uji berupa citra ikan segar sebanyak 14 citra ikan sebagai citra uji dan citra ikan tidak segar sebanyak 46 citra ikan sebagai citra latih, serta 2 jenis klasifikasi yaitu “ikan segar” dan “ikan tidak segar”. Berikut ini data lengkap dari citra ikan yang disajikan kedalam betuk tabel:

Tabel 4.9 Informasi Citra Latih dan Citra Uji

Data Citra	Jumlah	Klasifikasi	Jumlah
Citra Latih	46 Citra	Ikan Segar	23
		Ikan Tidak Segar	23
		Total	46
Citra Uji	14 Citra	Ikan Segar	7
		Ikan Tidak Segar	7
		Total	14

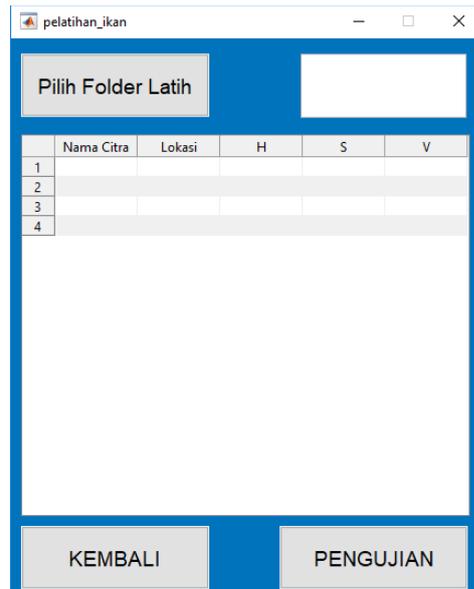
Dari tabel informasi citra ikan di atas, selanjutnya menyiapkan data citra ikan latih dan citra ikan uji yang disimpan kedalam folder “Data Latih” dan folder “Data Uji” berikut ini:

**Gambar 4.15** Citra Ikan Latih**Gambar 4.16** Citra Ikan Uji

Langkah selanjutnya ialah melakukan pelatihan untuk mendapatkan nilai HSV yang telah disimpan kedalam dataset citra ikan latih sebanyak 46 buah citra yang telah disiapkan pada gambar 4.15.

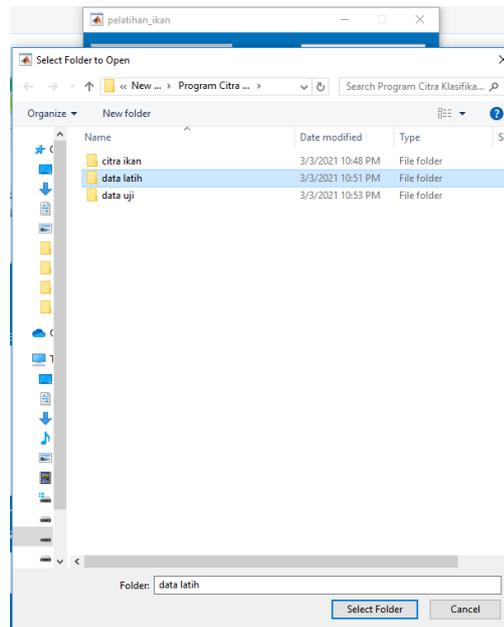
1. Proses Pelatihan Citra Ikan

Adapun tahapan pertama dalam melakukan klasifikasi kesegaran ikan adalah melakukan pelatihan agar data nilai citra ikan segar dan tidak segar tersimpan kedalam dataset. Adapun tahapannya pada sistem aplikasi adalah sebagai berikut:



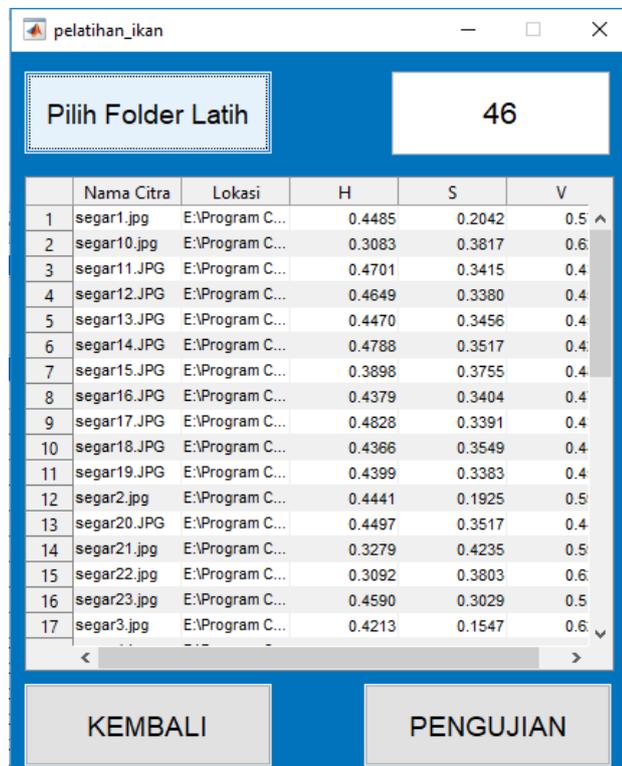
Gambar 4.17 Menu Proses Pelatihan

Dari gambar 4.17 diatas, ada beberapa *button* yang mempunyai fungsi masing-masing dan telah dijabarkan pada rancangan sebelumnya. Kemudian untuk melakukan proses pelatihan terlebih dahulu memilih folder tempat tersimpannya data citra ikan latih dengan cara menekan *button* “Pilih Folder Latih”, sehingga menampilkan menu pencarian seperti dibawah ini:



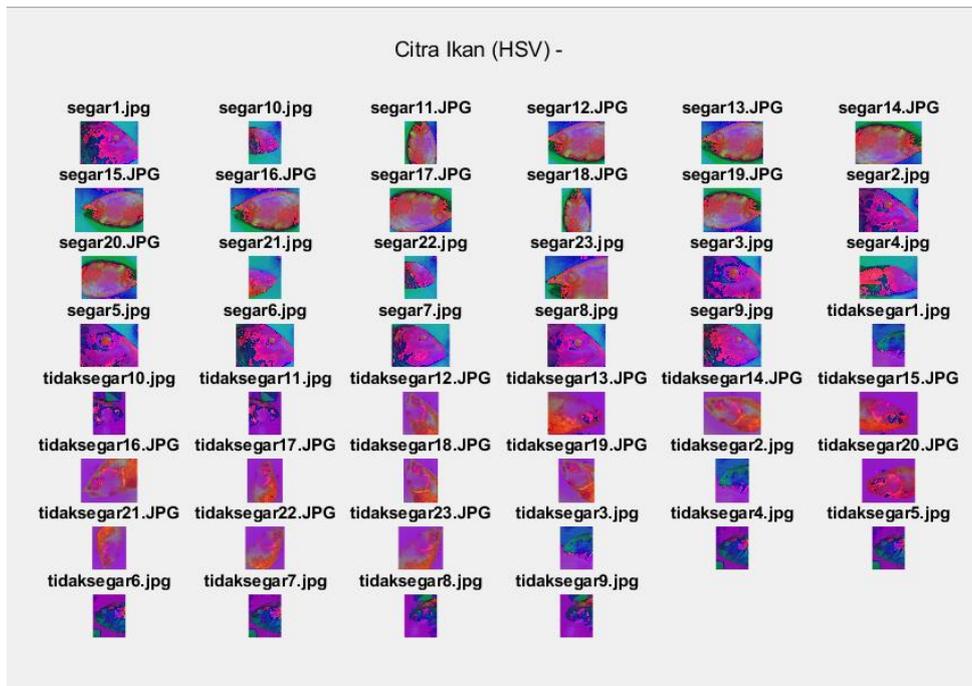
Gambar 4.18 Menu Proses Pencarian Folder Latih

Folder latih telah dipilih kemudian menekan *button* “*Select Folder*” dan akan menampilkan hasil pada aplikasi seperti berikut ini:



Gambar 4.19 Menu Proses Pelatihan Selesai

Pada gambar 4.19 diatas diketahui jumlah data latih sebanyak 46 citra ikan dengan nilai HSV yang terdapat pada tabel. Kemudian proses itu juga akan menampilkan citra HSV seperti berikut ini:



Gambar 4.20 Citra Latih Ikan HSV

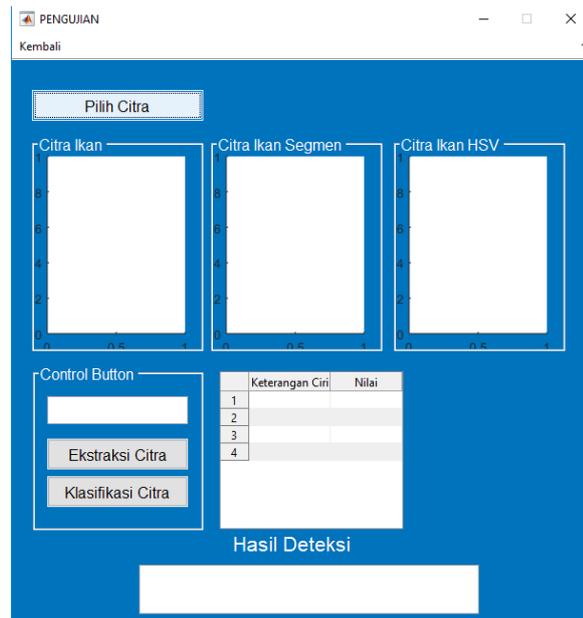
Setelah melakukan pelatihan maka muncullah figure untuk menampilkan isi keseluruhan folder yang akan dipilih dan dirubah kedalam bentuk citra HSV. Data nilai HSV tersebut akan disimpan kedalam dataset untuk proses selanjutnya yaitu pengujian klasifikasi kesegaran ikan.

2. Pengujian Kesegaran Ikan

Tahap pengujian ini dilakukan dengan cara menguji satu persatu. Pertama menguji 7 citra ikan dengan klasifikasi “ikan segar”, kemudian dilanjutkan dengan 7 citra ikan dengan klasifikasi “ikan tidak segar”. Hasil akhir adalah melakukan pengujian akurasi klasifikasi kesegaran citra ikan. Adapun tahapan pada aplikasi yang telah dibuat dapat dilihat sebagai berikut:

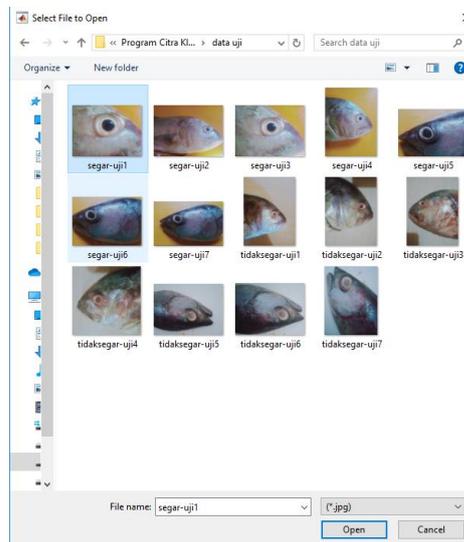
a. Klasifikasi Citra Ikan Kategori Segar

Berikut tahapan pada sistem aplikasi tersebut :



Gambar 4.21 Proses Pengujian Citra Ikan

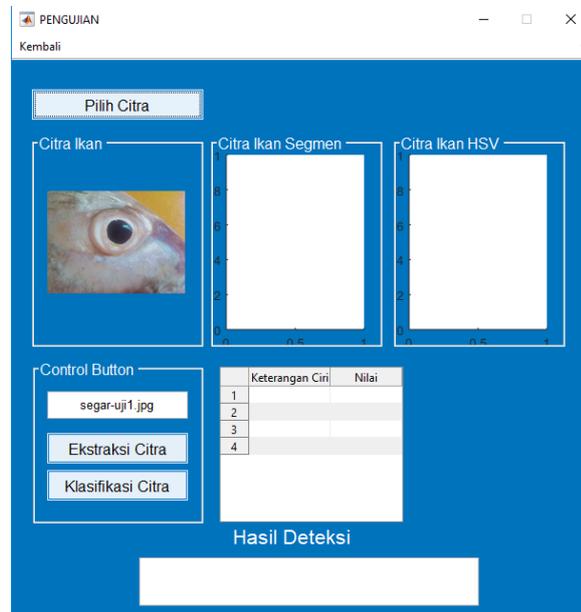
Gambar diatas merupakan proses pengujian citra yang kemudian akan diklasifikasi. Proses tersebut dimulai dengan mencari citra ikan uji dengan cara menekan *button* “Pilih Citra” sehingga akan menampilkan menu pencarian citra sebagai berikut:



Gambar 4.22 Menu Pencarian Citra

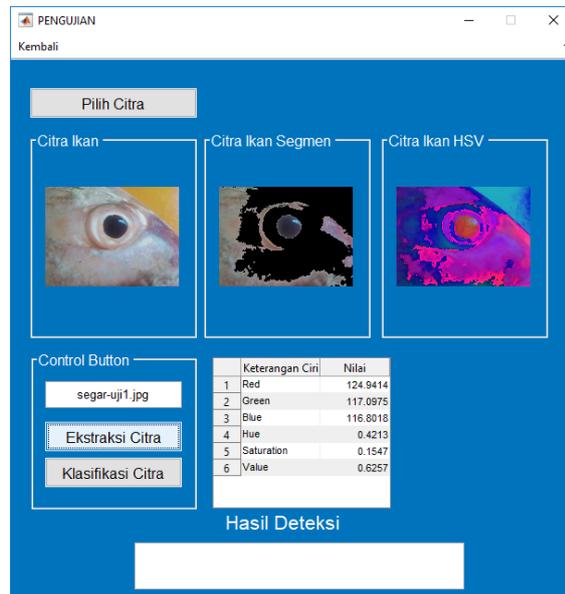
Gambar 4.22 diatas merupakan menu pencarian citra, pada menu tersebut terdapat 14 citra uji yang akan diuji satu persatu. Pengguna/user memilih citra uji yang akan

diklasifikasi jenis kesegarannya, dimulai dengan memilih citra “segar-1” lalu pilih “open”. Hasilnya seperti berikut ini:



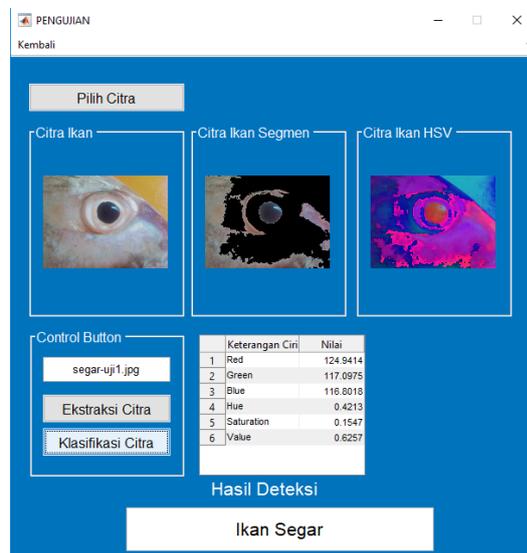
Gambar 4.23 Citra Ikan Segar 1 Dipilih

Setelah pemilihan citra ikan segar 1 yang kemudian akan diklasifikasi, sistem ini juga dapat membaca informasi nama citra tersebut. Kemudian untuk memulai proses selanjutnya yaitu proses klasifikasi kesegaran, pertama ialah melakukan ekstraksi citra ikan untuk mendapatkan nilai HSV nya dengan menekan button “Ekstraksi Citra” sehingga akan menampilkan citra ikan HSV seperti pada gambar berikut ini:



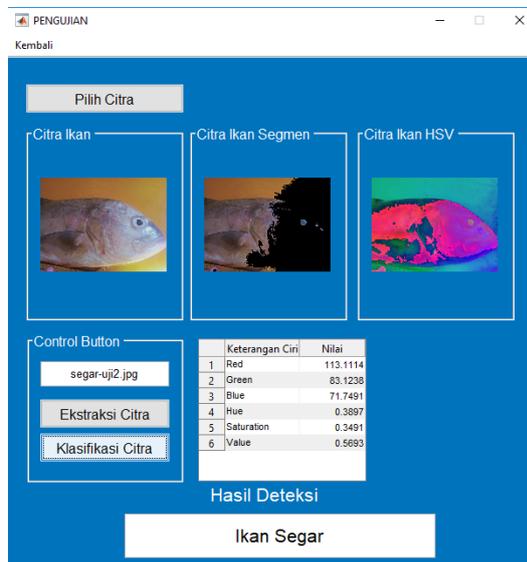
Gambar 4.24 Informasi Citra Ikan Segar 1

Setelah proses ekstraksi citra dilakukan dan nilai *HSV* sudah diketahui, selanjutnya melakukan klasifikasi kesegaran ikan dengan menekan *button* “Klasifikasi Citra” kemudian akan menampilkan hasil sebagai berikut:



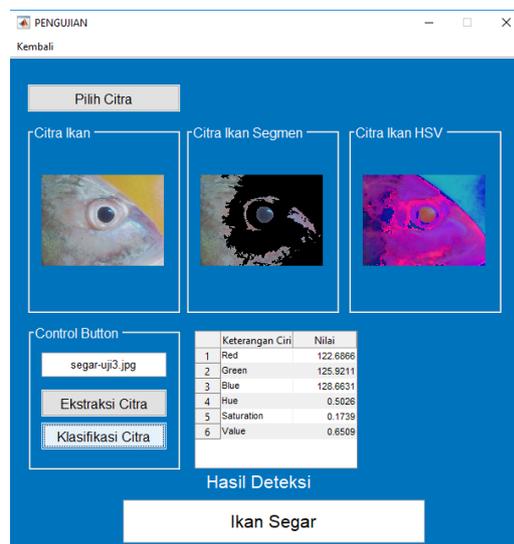
Gambar 4.25 Klasifikasi Ikan Segar 1

Proses klasifikasi citra ikan segar 1 berhasil diterapkan pada sistem ini, dimana citra uji ikan segar berhasil diklasifikasi sebagai “Ikan Segar”, sehingga klasifikasi sistem benar. Berikut adalah klasifikasi kesegaran ikan dengan data citra ikan segar 2 dengan hasil berikut ini:



Gambar 4.26 Klasifikasi Ikan Segar 2

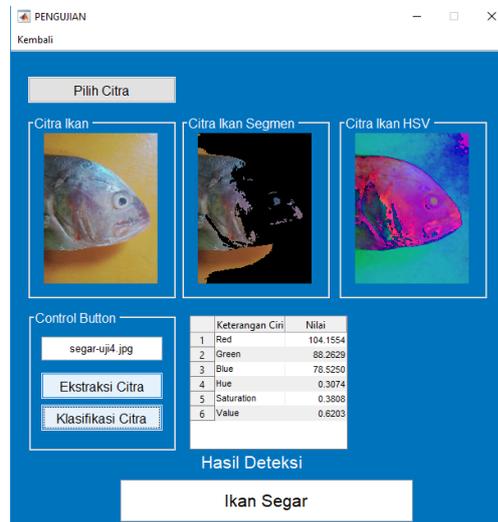
Proses klasifikasi ikan segar 2 berhasil diterapkan pada sistem aplikasi ini, dimana citra uji ikan segar berhasil diklasifikasi sebagai “Ikan Segar”, sehingga klasifikasi sistem benar. Berikut adalah klasifikasi kesegaran ikan dengan data citra ikan segar 3 dengan hasil berikut ini:



Gambar 4.27 Klasifikasi Ikan Segar 3

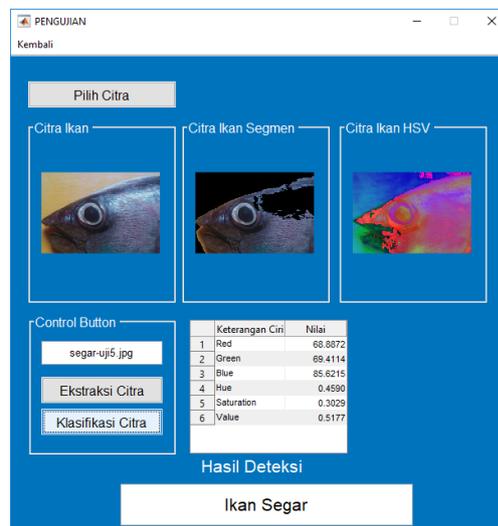
Proses klasifikasi ikan segar 3 juga berhasil diterapkan pada sistem aplikasi ini. Citra uji ikan segar 3 berhasil diklasifikasi sebagai “Ikan Segar”, sehingga

klasifikasi sistem benar. Berikut adalah klasifikasi kesegaran ikan dengan data citra ikan segar 4 dengan hasil berikut ini:



Gambar 4.28 Klasifikasi Ikan Segar 4

Proses klasifikasi ikan segar 4 berhasil diterapkan pada sistem ini, dimana citra uji ikan segar diklasifikasi sebagai “Ikan Segar”, sehingga klasifikasi sistem ini benar. Berikut adalah klasifikasi kesegaran ikan dengan data citra ikan segar 5 dengan hasil berikut ini:



Gambar 4.29 Klasifikasi Ikan Segar 5

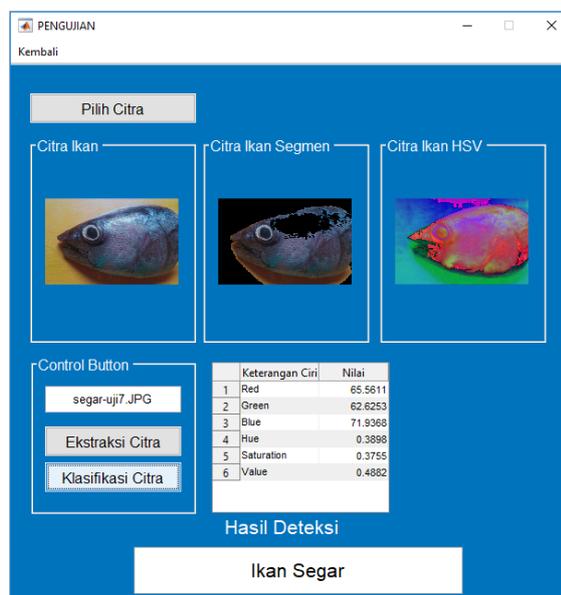
Dari gambar diatas, proses klasifikasi ikan segar 5 berhasil diterapkan pada sistem aplikasi, dengan citra uji ikan segar diklasifikasi sebagai “Ikan Segar”, sehingga

klasifikasi sistem ini benar. Berikut adalah klasifikasi kesegaran ikan dengan data citra uji ikan segar 6 dengan hasil berikut ini:



Gambar 4.30 Klasifikasi Ikan Segar 6

Proses klasifikasi ikan segar 6 juga berhasil diterapkan pada sistem aplikasi dimana dengan citra uji ikan segar diklasifikasi sebagai “Ikan Segar”, sehingga klasifikasi sistem aplikasi ini benar. Berikut adalah klasifikasi kesegaran ikan dengan data citra ikan segar 7 dengan hasil berikut ini:



Gambar 4.31 Klasifikasi Ikan Segar 7

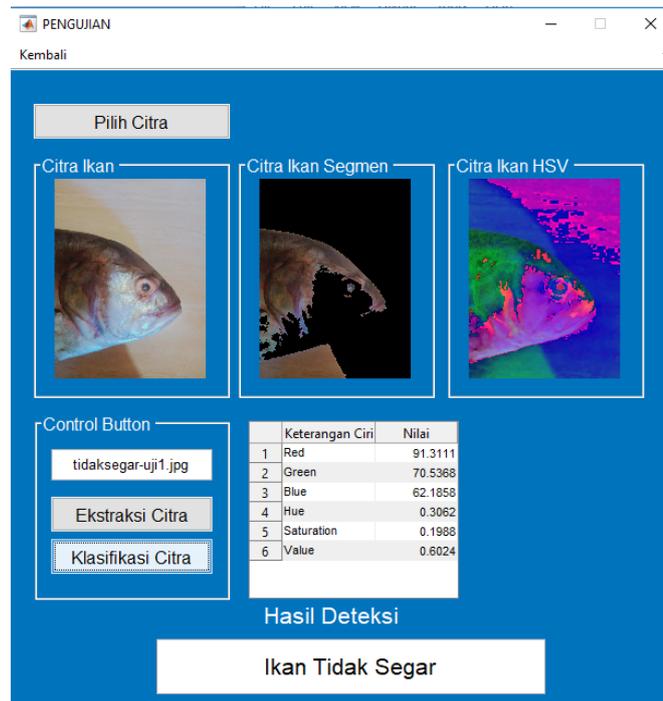
Dari gambar klasifikasi diatas, proses klasifikasi ikan segar 7 juga berhasil diterapkan di sistem aplikasi ini, dengan data citra uji ikan segar diklasifikasi sebagai “Ikan Segar”, sehingga klasifikasi sistem benar. Berikut adalah hasil keseluruhan uji klasifikasi kesegaran ikan kategori “ikan segar” dengan menguji sampel citra sebanyak 7 citra uji.

Tabel 4.10 Hasil Pengujian Citra Ikan Segar

No	Nama Citra	Klasifikasi	Hasil Klasifikasi	Keterangan
1	Segar-uji1.jpg	Ikan Segar	Ikan Segar	Benar
2	Segar-uji2.jpg	Ikan Segar	Ikan Segar	Benar
3	Segar-uji3.jpg	Ikan Segar	Ikan Segar	Benar
4	Segar-uji4.jpg	Ikan Segar	Ikan Segar	Benar
5	Segar-uji5.jpg	Ikan Segar	Ikan Segar	Benar
6	Segar-uji6.jpg	Ikan Segar	Ikan Segar	Benar
7	Segar-uji7.jpg	Ikan Segar	Ikan Segar	Benar

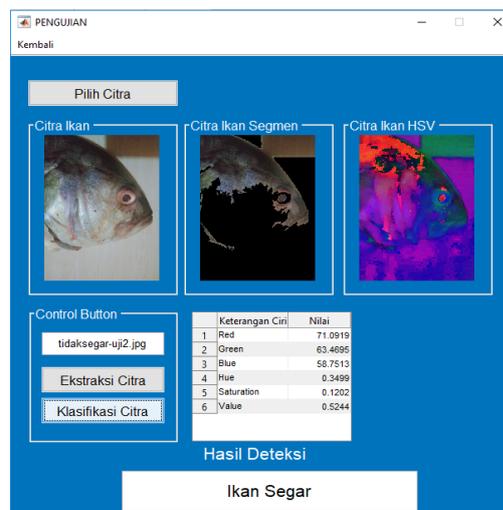
b. Klasifikasi Citra Ikan Kategori Tidak Segar

Proses klasifikasi citra ikan kategori tidak segar ini sama dengan proses sebelumnya. Hasil akhir dari klasifikasi kesegaran untuk data citra ikan tidak segar ialah sebagai berikut:



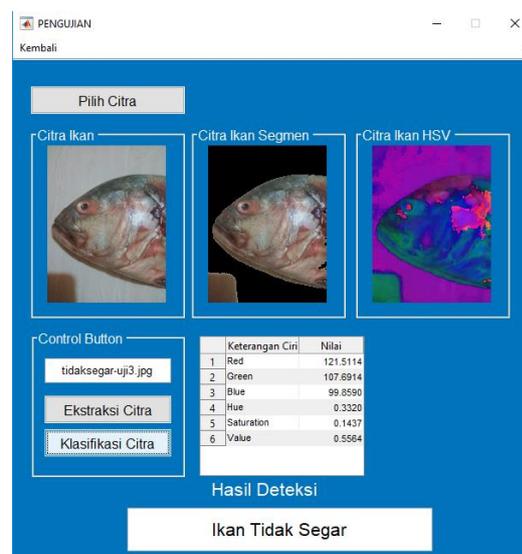
Gambar 4.32 Klasifikasi Ikan Tidak Segar 1

Gambar diatas merupakan proses klasifikasi untuk kategori citra ikan tidak segar. Proses tersebut sama seperti pada klasifikasi ikan segar sebelumnya, dan citra uji juga sebanyak 7 sampel citra uji ikan tidak segar. Klasifikasi citra ikan tidak segar 1 berhasil diterapkan di sistem aplikasi ini, dimana citra uji ikan tidak segar berhasil diklasifikasi sebagai "Ikan Tidak Segar", maka sistem klasifikasi benar. Berikut adalah klasifikasi kesegaran data citra ikan tidak segar 2 hasilnya sebagai berikut:



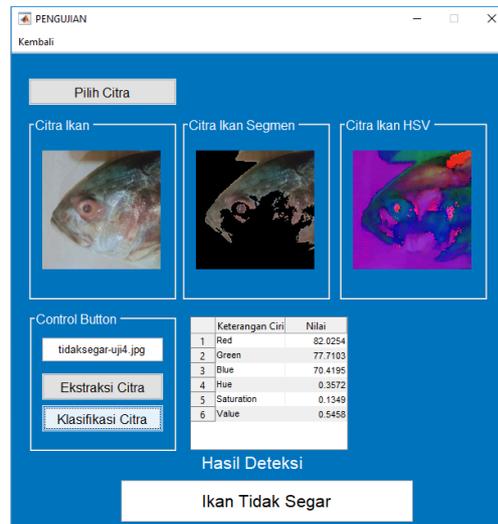
Gambar 4.33 Klasifikasi Ikan Tidak Segar 2

Dari gambar diatas, proses klasifikasi ikan tidak segar 2 ternyata tidak berhasil diterapkan oleh sistem tersebut, dengan data citra uji ikan tidak segar tidak berhasil diklasifikasi sebagai “Ikan Tidak Segar”, sehingga klasifikasi sistem salah. Berikut adalah klasifikasi kesegaran ikan untuk data citra ikan tidak segar 3 dengan hasil berikut ini:



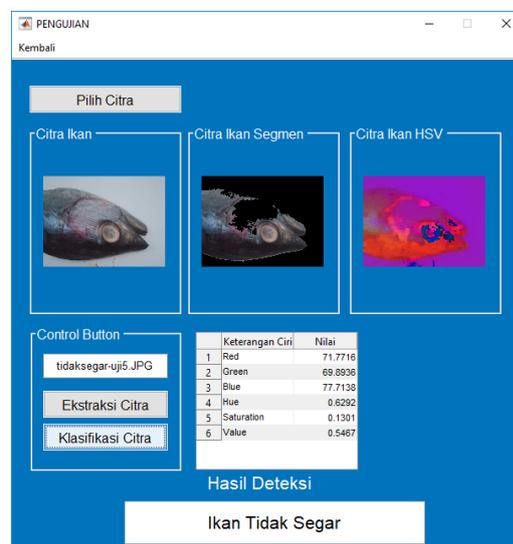
Gambar 4.34 Klasifikasi Ikan Tidak Segar 3

Kemudian untuk proses klasifikasi ikan tidak segar 3 berhasil diterapkan di sistem aplikasi ini, dengan citra uji ikan tidak segar diklasifikasi sebagai “Ikan Tidak Segar”, sehingga klasifikasi sistem benar. Berikut adalah klasifikasi kesegaran ikan dengan data citra ikan tidak segar 4 dengan hasil berikut ini:



Gambar 4.35 Klasifikasi Ikan Tidak Segar 4

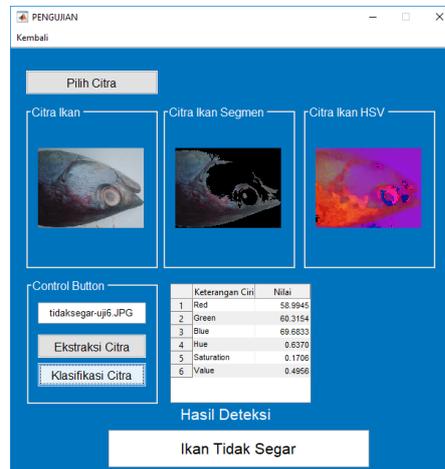
Selanjutnya untuk proses klasifikasi ikan tidak segar 4 berhasil diterapkan pada sistem aplikasi ini, dengan data citra uji tidak segar berhasil diklasifikasi sebagai “Ikan Tidak Segar”, sehingga klasifikasi sistem benar. Berikut adalah klasifikasi kesegaran ikan dengan data citra ikan tidak segar 5 dengan hasil berikut ini:



Gambar 4.36 Klasifikasi Ikan Tidak Segar 5

Dari gambar diatas, proses klasifikasi ikan tidak segar 5 berhasil diterapkan di sistem aplikasi, dengan data citra uji ikan tidak segar berhasil diklasifikasi sebagai

“Ikan Tidak Segar”, maka klasifikasi sistem benar. Berikut adalah klasifikasi kesegaran ikan dengan data citra ikan tidak segar 6 dengan hasil berikut ini:



Gambar 4.37 Klasifikasi Ikan Tidak Segar 6

Kemudian untuk proses klasifikasi ikan tidak segar 6 juga berhasil diterapkan di sistem aplikasi ini, data citra uji ikan tidak segar berhasil diklasifikasi sebagai “Ikan Tidak Segar”, maka klasifikasi sistem benar. Berikut adalah klasifikasi kesegaran ikan dengan data citra ikan tidak segar 7 dengan hasil berikut ini:



Gambar 4.38 Klasifikasi Ikan Tidak Segar 7

Selanjutnya untuk proses klasifikasi terakhir ini yaitu citra ikan tidak segar 7 juga berhasil diterapkan di sistem aplikasinya, dengan data citra uji ikan tidak segar

berhasil diklasifikasi sebagai “Ikan Tidak Segar”, maka klasifikasi sistem ini benar. Berikut adalah hasil keseluruhan uji klasifikasi kesegaran ikan kategori “ikan tidak segar” dengan menguji sebanyak 7 sampel citra:

Tabel 4.11 Hasil Pengujian Citra Ikan Tidak Segar

No	Nama Citra	Klasifikasi	Hasil Klasifikasi	Keterangan
1	Tidaksegar-uji1.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar
2	Tidaksegar-uji2.jpg	Ikan Tidak Segar	Ikan Segar	Salah
3	Tidaksegar-uji3.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar
4	Tidaksegar-uji4.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar
5	Tidaksegar-uji5.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar
6	Tidaksegar-uji6.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar
7	Tidaksegar-uji7.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar

Berdasarkan dari hasil pengujian klasifikasi kesegaran ikan dengan 46 citra ikan kategori segar dan tidak segar maka hasil klasifikasinya adalah sebagai berikut:

Tabel 4.12 Hasil Pengujian Klasifikasi Kesegaran Citra Ikan Keseluruhan

No	Nama Citra	Klasifikasi	Hasil Klasifikasi	Keterangan
1	Segar-uji1.jpg	Ikan Segar	Ikan Segar	Benar
2	Segar-uji2.jpg	Ikan Segar	Ikan Segar	Benar
3	Segar-uji3.jpg	Ikan Segar	Ikan Segar	Benar
4	Segar-uji4.jpg	Ikan Segar	Ikan Segar	Benar
5	Segar-uji5.jpg	Ikan Segar	Ikan Segar	Benar
6	Segar-uji6.jpg	Ikan Segar	Ikan Segar	Benar

Tabel Lanjutan 4.12 Hasil Pengujian Klasifikasi Kesegaran Citra Ikan
Keseluruhan

No	Nama Citra	Klasifikasi	Hasil Klasifikasi	Keterangan
7	Segar-uji7.jpg	Ikan Segar	Ikan Segar	Benar
8	Tidaksegar-uji1.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar
9	Tidaksegar-uji2.jpg	Ikan Tidak Segar	Ikan Segar	Salah
10	Tidaksegar-uji3.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar
11	Tidaksegar-uji4.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar
12	Tidaksegar-uji5.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar
13	Tidaksegar-uji6.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar
14	Tidaksegar-uji7.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar

Berdasarkan pada tabel hasil pengujian diatas, didapati citra uji dengan klasifikasi benar dan klasifikasi salah, dimana terdapat 14 citra ikan uji dengan hasil klasifikasi yang benar sebanyak 13 citra dan klasifikasi yang salah sebanyak 1 citra ikan. Kemudian menghitung tingkat akurasi berdasarkan hasil uji dari citra ikan sebanyak 14 citra, dengan menggunakan rumus sebagai berikut:

$$Akurasi = \frac{Jumlah\ Klasifikasi\ Benar}{Jumlah\ Data} \times 100\%$$

$$Akurasi = \frac{13}{14} \times 100\% = 93\%$$

Berdasarkan dari hasil uji akurasi diatas, didapatkan nilai akurasi sebesar 93% untuk proses klasifikasi kesegaran ikan dengan menggunakan sampel uji sebanyak 14 citra

4.2.2. Penerapan

Penerapan/penggunaan sistem ini digunakan untuk mengetahui tingkat kesegaran pada ikan. Dengan menggunakan ekstraksi fitur warna Hue Saturation Value dan menerapkan metode K-Nearest Neighbor untuk mengklasifikasi kesegaran ikan dalam bentuk citra berguna untuk membantu mahasiswa maupun

pelajar yang ingin mengetahui pengolahan citra tersebut kedalam bentuk citra digital. Penerapan sistem ini dapat diterapkan oleh mahasiswa ataupun pelajar dimana sistem ini akan mempermudah menganalisis tingkat kesegaran pada ikan dan meminimalisir kesalahan dalam mengetahui tingkat kesegarannya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian dan hasil pengujian yang dilakukan dengan klasifikasi kesegaran ikan, dapat diambil kesimpulan sebagai berikut:

1. Ekstraksi menggunakan metode ruang warna *HSV* dapat dilakukan pada citra ikan dengan bentuk segar dan tidak segar
2. Klasifikasi kesegaran ikan dengan 14 data uji memberikan hasil akurasi sebesar 93% dengan 13 data uji benar dan 1 data uji salah diklasifikasi.
3. Klasifikasi gagal dilakukan dengan citra ikan yang memiliki jarak pengambilan terlalu jauh.
4. Dari hasil pengujian yang didapat, akurasi klasifikasi tentu akan berubah jika data yang diuji semakin banyak

5.2 Saran

Dari hasil penelitian yang didapat masih terdapat kesalahan, untuk pengembangan sistem yang lebih baik pada penelitian selanjutnya maka peneliti memberikan saran sebagai berikut:

1. Untuk pengembangan sistem selanjutnya perlu adanya perbandingan metode ekstraksi *HSV* dengan metode ekstraksi lainnya sehingga memberikan tingkat klasifikasi yang lebih akurat
2. Perlu adanya penambahan metode yang dapat mendeteksi sisi-sisi citra ikan sehingga proses klasifikasi lebih baik dan akurat.
3. Untuk mendapatkan hasil klasifikasi yang lebih optimal perlu ditambahkan data uji dan data latih yang lebih banyak
4. Perlu ditambahkan menu pengujian dengan pemilihan folder citra ikan secara klasifikasi bersamaan, sehingga dapat mempersingkat waktu pengujian jika citra uji cukup banyak

DAFTAR PUSTAKA

- Adawyah, R. 2007. Pengolahan dan Pengawetan Ikan. Jakarta: Bumi Aksara.
- Agaputra, D, M., Wardani, R, R, K., Siswanto, E., 2013. “ Pencarian Citra Digital Berbasis Konten dengan Ekstraksi Fitur HSV, ACD, dan GLCM “. Institut Teknologi Harapan Bangsa, Telematika, Vol.8, No. 2.
- D. Syahid, Jumadi, and D. Nursantika, “Sistem Klasifikasi Jenis Tanaman Hias Daun Philodendron Menggunakan Metode K-Nearest Neighbor (knn) Berdasarkan Nilai Hue, Saturation, Value (hsv)”, *J. Online Inform.*, vol. 1, pp.20-23, 2016
- E. Budianita, J. Jasril, and L. Handayani, “Implementasi Pengolahan Citra dan Klasifikasi K-Nearest Neighbour Untuk Membangun Aplikasi Pembeda Daging Sapi dan Babi Berbasis Web”, *J. Sains dan Teknol. Ind.*, vol. 12, no. Vol 12, No 2 (2015): Juni 2015, pp. 242-247, 2015.
- Eko, H. & Abu Salam. “Pengukuran Tingkat Kematangan Kopi Robusta Menggunakan Algoritma K-Nearest Neighbor”. Universitas Dian Nuswantoro, Semarang. 2018
- Faris Mushlihul Amin, 2018. Identifikasi Citra Daging Ayam Berformalin Menggunakan Metode Fitur Tekstur Dan K-Nearest Neighbor (k-nn).
- Food and Agriculture Organization [FAO]. (1995). *Quality and Quality Changes in Fresh Fish*. Huss HH, editor. FAO
- Furqan,M., Sriani., Y.K., 2020 Perbandingan Algoritma *Contraharmonic Mean Filter Dan Arithmetic Mean Filter* Untuk Mereduksi *Exponential Noise*. *JISKA*, Vol.5, 107-115
- Furqan,M., Sriani., Sari,I.E.Y., 2020, Penerapan Metode Otsu Dalam Melakukan Segmentasi Citra Pada Citra Naskah Arab. *Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, Vol.20, 59-72
- Furqan,M., Sriani., Harahap,L.S., 2020, Klasifikasi Daun Bugenvil Menggunakan *Gray Level Co-Occurrence Matrix dan K-Nearest Neighbor*. *Jurnal CoreIT*, Vol.6 No.1,pp. 2-29, 2020

Gonzales, R.C., and Woods, R.E. 2002. *Digital Image Processing* Second edition. New Jersey: *Prentice Hall*

Hanselman D, L. (1998). *Mastering Matlab 5, A Comprehensive Tutorial and Reference. Prentice-Hall Inc.*

Irawan, A. 1995, *Pengawetan Ikan dan Hasil Perikanan. Cara Mengolah dan Mengawetkan secara Tradisional dan Modern*, CV. Aneka, Solo.

Junianto, 2003. *Teknik Penanganan Ikan*. Swadaya, Jakarta.

Kadir, A. & Susanto, A., 2012. *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Andi.

McAndrew. A. (2004). *An Introduction to Digital Image Processing with Matlab. Australia. Thomson.*

Putra, Darma. "Dasar Pengolahan Citra" *Pengolahan Citra*, Ed.1. Yogyakarta: ANDI, 2010, BAB 3, 2.8.4, pp.57.

Prasetyo, E., 2011. *Pengolahan Citra Digital Dan Aplikasinya Menggunakan Matlab*. Yogyakarta: Penerbit Andi.

T. Sutoyo, E. Mulyanto, V. Suhartono, O. D. Nurhayati and W., *Teori Pengolahan Citra Digital*, ANDI, 2009.

T. W. Adi P, "Pengenalan Wajah Dengan Matriks Kookurensi Aras Keabuan dan Jaringan Syaraf Tiruan Probabilistik," in *UNIVERSITAS DIPONEGORO*, 2013.

Winarni, T., F. Swastawati., Y.S. Darmanto., E.N. Dewi. 2003. *Uji mutu terpadu pada beberapa spesies ikan dan produk perikanan di Indonesia. Laporan Akhir Hibah Bersaing XI Perguruan Tinggi. Universitas Diponegoro, Semarang.*

Z. Dwi Lestari, N. Nafi'iyah, and P. Hadisusilo, "Sistem Klasifikasi Jenis Pisang Berdasarkan Ciri Warna HSV Menggunakan Metode K- Nearest Neighbor", *Seminar Nasional Teknologi Info.. dan Komunikasi 2019, Tekologi Humanis di Era Society 5.00*

LAMPIRAN-LAMPIRAN

Kode program matlab untuk klasifikasi kesegaran ikan berdasarkan ekstraksi fitur menggunakan metode K-Nearest Neighbor dan Hue Saturation Value

```
function varargout = menu_utama(varargin)
% MENU_UTAMA MATLAB code for menu_utama.fig
%     MENU_UTAMA, by itself, creates a new MENU_UTAMA or raises
the existing
%     singleton*.
%
%     H = MENU_UTAMA returns the handle to a new MENU_UTAMA or the
handle to
%     the existing singleton*.
%
%     MENU_UTAMA('CALLBACK',hObject,eventData,handles,...) calls
the local
%     function named CALLBACK in MENU_UTAMA.M with the given input
arguments.
%
%     MENU_UTAMA('Property','Value',...) creates a new MENU_UTAMA
or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before menu_utama_OpeningFcn gets called.
An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to menu_utama_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help menu_utama

% Last Modified by GUIDE v2.5 23-Feb-2021 21:24:07

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @menu_utama_OpeningFcn, ...
                  'gui_OutputFcn',  @menu_utama_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
% End initialization code - DO NOT EDIT

% --- Executes just before menu_utama is made visible.
function menu_utama_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to menu_utama (see VARARGIN)

% Choose default command line output for menu_utama
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes menu_utama wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = menu_utama_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

close;
guidata(pelatihan_ikan);
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close;
guidata(pengujian_kesegaran_ikan);

```

```

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close;
guidata(tentang_penulis);

                                PELATIHAN_IKAN
function varargout = pelatihan_ikan(varargin)
% PELATIHAN_IKAN MATLAB code for pelatihan_ikan.fig
%     PELATIHAN_IKAN, by itself, creates a new PELATIHAN_IKAN or
raises the existing
%     singleton*.
%
%     H = PELATIHAN_IKAN returns the handle to a new PELATIHAN_IKAN
or the handle to
%     the existing singleton*.
%
%     PELATIHAN_IKAN('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in PELATIHAN_IKAN.M with the given
input arguments.
%
%     PELATIHAN_IKAN('Property','Value',...) creates a new
PELATIHAN_IKAN or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before pelatihan_ikan_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to pelatihan_ikan_OpeningFcn
via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help pelatihan_ikan

% Last Modified by GUIDE v2.5 04-Mar-2021 00:50:27

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @pelatihan_ikan_OpeningFcn,
                  ...
                  'gui_OutputFcn',    @pelatihan_ikan_OutputFcn,
                  ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',     []);

```

```

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

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before pelatihan_ikan is made visible.
function pelatihan_ikan_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to pelatihan_ikan (see VARARGIN)

% Choose default command line output for pelatihan_ikan
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes pelatihan_ikan wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = pelatihan_ikan_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

folder = uigetdir(); %menampilkan jendela open folder untuk memilik
folder dimana citra berada
d = dir([folder, '*.*jpg']); %mengakses data citra png yang terdapat
pada folder yang dipilih

```

```

n=length(d); %menghitung jumlah file yang png yang terdapat pada
direktori

citra=cell(n,1); %menciptakan cell nx1 untuk menyimpan data citra
yang terdapat pada direktori yang dipilih
lokasi=cell(n,1); %menciptakan cell nx1 untuk menyimpan lokasi path
citra yang terdapat pada direktori yang dipilih
H=cell(n,1);%menciptakan cell nx1 untuk menyimpan rata-rata nilai
H dari citra yang dibaca
S=cell(n,1);%menciptakan cell nx1 untuk menyimpan rata-rata nilai
S dari citra yang dibaca
V=cell(n,1);%menciptakan cell nx1 untuk menyimpan rata-rata nilai
V dari citra yang dibaca
data=get(handles.tblDataCitraLatih,'Data');%mengambil data pada
tabel citra latih
x=isempty(data{1});

%menyimpan data nama citra,kriteria,dan target keluaran ke dalam
cell dan
%menampilkan citra latih RGB
figure
n_row=fix(n/5);
n_col=5+mod(n,5);
for i=1:n
    citra{i,1}=d(i).name;
    lokasi{i,1}=folder;
    %membaca citra
    path=strcat(folder,'\ ',d(i).name);
    img=imread(path);

    %menampilkan citra
    subplot(n_row,n_col,i),imshow(img),title(d(i).name);
end
suptitle(strcat('Citra Ikan (RGB)',' - '))

%menyimpan data nama citra,kriteria,dan target keluaran ke dalam
cell dan
%menampilkan citra latih HSV
figure
n_row=fix(n/5);
n_col=5+mod(n,5);
for i=1:n
    citra{i,1}=d(i).name;
    lokasi{i,1}=folder;
    %membaca citra
    path=strcat(folder,'\ ',d(i).name);
    img=imread(path);
    hsv=rgb2hsv(img);
    H{i,1}=mean(mean(hsv(:, :, 1)));
    S{i,1}=mean(mean(hsv(:, :, 2)));
    V{i,1}=mean(mean(hsv(:, :, 3)));

    %menampilkan citra
    subplot(n_row,n_col,i),imshow(hsv),title(d(i).name);
end

```

```

suptitle(strcat('Citra Ikan (HSV)', ' - '))

datacitra=[citra,lokasi,H,S,V];
data_hsv=[cell2mat(H),cell2mat(S),cell2mat(V)];
total=0;
if x==1
    datacitra=[citra,lokasi,H,S,V];
    total=0+n;
else
    datacitra=cat(0,data,datacitra);
    total=size(data,1)+n;
end
%menampilkan data citra,kriteria,dan target pada tabel data citra
latih
set(handles.tblDataCitraLatih,'Data',datacitra);
%menampilkan total citra
set(handles.edit1,'String',num2str(total));

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1
%        as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%        called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close;
guidata(pengujian_kesegaran_ikan);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)
close;
guidata(menu_utama);

                                PENGUJIAN_KESEGARAN_IKAN
function varargout = penguajian_kesegaran_ikan(varargin)
%      PENGUJIAN_KESEGARAN_IKAN      MATLAB      code      for
penguajian_kesegaran_ikan.fig
%      PENGUJIAN_KESEGARAN_IKAN, by itself, creates a new
PENGUJIAN_KESEGARAN_IKAN or raises the existing
%      singleton*.
%
%      H = PENGUJIAN_KESEGARAN_IKAN returns the handle to a new
PENGUJIAN_KESEGARAN_IKAN or the handle to
%      the existing singleton*.
%
%
%      PENGUJIAN_KESEGARAN_IKAN('CALLBACK', hObject, eventData, handles,...)
calls the local
%      function named CALLBACK in PENGUJIAN_KESEGARAN_IKAN.M with
the given input arguments.
%
%      PENGUJIAN_KESEGARAN_IKAN('Property','Value',...) creates a
new PENGUJIAN_KESEGARAN_IKAN or raises the
%      existing singleton*. Starting from the left, property value
pairs are
%      applied to the GUI before penguajian_kesegaran_ikan_OpeningFcn
gets called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to
penguajian_kesegaran_ikan_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
penguajian_kesegaran_ikan

% Last Modified by GUIDE v2.5 04-Mar-2021 01:00:24

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',      mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn',
@penguajian_kesegaran_ikan_OpeningFcn, ...
                  'gui_OutputFcn',
@penguajian_kesegaran_ikan_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})

```

```

    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before pengujian_kesegaran_ikan is made visible.
function pengujian_kesegaran_ikan_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to pengujian_kesegaran_ikan (see
VARARGIN)

% Choose default command line output for pengujian_kesegaran_ikan
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes pengujian_kesegaran_ikan wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = pengujian_kesegaran_ikan_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% menampilkan menu open file
[nama_file, nama_path] = uigetfile('*.jpg');
if ~isequal(nama_file,0)
    % membaca file citra
    Img = imread(fullfile(nama_path, nama_file));

```

```

    % menampilkan citra pada axes 1
    axes(handles.axes1)
    imshow(Img)
    % menampilkan nama file citra pada edit1
    set(handles.edit1,'String',nama_file)
    % menyimpan variabel Img pada lokasi handles
    handles.Img = Img;
    guidata(hObject, handles)
else
    % jika tidak ada file yang dipilih maka akan kembali
    return
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2
as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1
as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if      ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% memanggil variabel Img yang ada di lokasi handles
Img = handles.Img;
Img_gray = rgb2gray(Img);
bw = im2bw(Img_gray,graythresh(Img_gray));
bw = imcomplement(bw);
bw = imfill(bw,'holes');
bw = bwareaopen(bw,100);

% ekstraksi komponen RGB
R = Img(:,:,1);
G = Img(:,:,2);
B = Img(:,:,3);

% mengubah nilai background menjadi nol
R(~bw) = 0;
G(~bw) = 0;
B(~bw) = 0;
RGB = cat(3,R,G,B);
axes(handles.axes2)
imshow(RGB)
% menyimpan variabel bw pada lokasi handles
handles.bw = bw;
guidata(hObject, handles)
Red = sum(sum(R))/sum(sum(bw));
Green = sum(sum(G))/sum(sum(bw));
Blue = sum(sum(B))/sum(sum(bw));

% ekstraksi ciri warna HSV
HSV = rgb2hsv(Img);
H = HSV(:,:,1);
S = HSV(:,:,2);
V = HSV(:,:,3);
Hue = mean(mean(H));
Saturation = mean(mean(S));
Value = mean(mean(V));
axes(handles.axes3)
imshow(HSV)
Area = sum(sum(bw));

```

```

% Menambahkan hasil ekstraksi nilai RGB dan nilai HSV citra ikan
pada
% tabel
nilai_Ikan = cell(6,2);
nilai_Ikan{1,1} = 'Red';
nilai_Ikan{2,1} = 'Green';
nilai_Ikan{3,1} = 'Blue';
nilai_Ikan{4,1} = 'Hue';
nilai_Ikan{5,1} = 'Saturation';
nilai_Ikan{6,1} = 'Value';
nilai_Ikan{1,2} = Red;
nilai_Ikan{2,2} = Green;
nilai_Ikan{3,2} = Blue;
nilai_Ikan{4,2} = Hue;
nilai_Ikan{5,2} = Saturation;
nilai_Ikan{6,2} = Value;

% menampilkan ciri_ikan pada tabel
set(handles.uitable1,'Data',nilai_Ikan,'RowName',1:6)
% menampilkan ciri_ikan pada tabel
ciri_uji = [Red,Green,Blue,Hue,Saturation,Value, Area];
% menyimpan variabel ciri_uji pada lokasi handles
handles.ciri_uji = ciri_uji;
guidata(hObject, handles)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% memanggil variabel ciri_uji pada lokasi handles
ciri_uji = handles.ciri_uji;
% load hasil pelatihan dataset
load hasil_pelatihan
% standarisasi ciri uji
ciri_ujiZ = (ciri_uji - muZ)./sigmaZ;
score_uji = ciri_ujiZ*coeff;
HSV1 = score_uji(:,1);
HSV2 = score_uji(:,2);

% mengujikan data uji pada dataset
hasil_uji = predict(Mdl,[HSV1,HSV2]);
% menampilkan hasil pengujian pada edit2
set(handles.edit2,'String',hasil_uji)

% -----
-----
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject      handle to Untitled_1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
close;
guidata(menu_utama);

```

```

                                TENTANG_PENULIS
function varargout = tentang_penulis(varargin)
% TENTANG_PENULIS MATLAB code for tentang_penulis.fig
%     TENTANG_PENULIS, by itself, creates a new TENTANG_PENULIS or
%     raises the existing
%     singleton*.
%
%     H = TENTANG_PENULIS returns the handle to a new
TENTANG_PENULIS or the handle to
%     the existing singleton*.
%
%     TENTANG_PENULIS('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in TENTANG_PENULIS.M with the given
input arguments.
%
%     TENTANG_PENULIS('Property','Value',...) creates a new
TENTANG_PENULIS or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before tentang_penulis_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to tentang_penulis_OpeningFcn
via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help tentang_penulis

% Last Modified by GUIDE v2.5 04-Mar-2021 01:56:16

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @tentang_penulis_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @tentang_penulis_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
% End initialization code - DO NOT EDIT

% --- Executes just before tentang_penulis is made visible.
function tentang_penulis_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to tentang_penulis (see
VARARGIN)

% Choose default command line output for tentang_penulis
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes tentang_penulis wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = tentang_penulis_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on slider movement.
function slider1_Callback(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine
range of slider

% --- Executes during object creation, after setting all properties.
function slider1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

```

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close;
guidata(menu_utama);

                                DETEKSI

clc; clear; close all;

% membaca file citra
nama_folder = 'data latihan';
nama_file = dir(fullfile(nama_folder, '*.jpg'));
jumlah_file = numel(nama_file);
% inialisasi variabel ciri_latih
ciri_latih = zeros(jumlah_file,7);

for n = 1:jumlah_file
    % membaca citra RGB
    Img = imread(fullfile(nama_folder,nama_file(n).name));
    Img_gray = rgb2gray(Img);
    bw = im2bw(Img_gray,graythresh(Img_gray));
    bw = imcomplement(bw);
    bw = imfill(bw,'holes');
    bw = bwareaopen(bw,100);

    % ekstraksi ciri warna RGB
    R = Img(:,:,1);
    G = Img(:,:,2);
    B = Img(:,:,3);
    R(~bw) = 0;
    G(~bw) = 0;
    B(~bw) = 0;
    Red = sum(sum(R))/sum(sum(bw));
    Green = sum(sum(G))/sum(sum(bw));
    Blue = sum(sum(B))/sum(sum(bw));

    % ekstraksi ciri warna HSV
    HSV = rgb2hsv(Img);
    H = HSV(:,:,1);
    S = HSV(:,:,2);
    V = HSV(:,:,3);
    Hue = mean(mean(H));
    Saturation = mean(mean(S));
    Value = mean(mean(V));

```

```

    % ekstraksi ciri ukuran
Area = sum(sum(bw));
    % menambahkan nilai ekstraksi HSV pada variabel ciri_latih
    ciri_latih(n,1) = Red;
    ciri_latih(n,2) = Green;
    ciri_latih(n,3) = Blue;
    ciri_latih(n,4) = Hue;
    ciri_latih(n,5) = Saturation;
    ciri_latih(n,6) = Value;
    ciri_latih(n,7) = Area;
end

% standarisasi data
[ciri_latihZ,muZ,sigmaZ] = zscore(ciri_latih);
[coeff,score_latih,latent,tsquared,explained] = pca(ciri_latihZ);

% inisialisasi variabel kelas_latih
kelas_latih = cell(jumlah_file,1);
% mengisi nama deteksi buah ikan pada variabel kelas_latih
for k = 1:23
    kelas_latih{k} = 'Ikan Segar';
end

for k = 24:46
    kelas_latih{k} = 'Ikan Tidak Segar';
end

% ekstrak HSV
HSV1 = score_latih(:,1);
HSV2 = score_latih(:,2);
x1 = HSV1(1:23);
x2 = HSV1(24:46);
y2 = HSV2(24:46);

% klasifikasi menggunakan jatrak
Mdl = fitcknn([HSV1,HSV2],kelas_latih,'NumNeighbors',2);

% menyimpan variabel-variabel hasil pelatihan
save hasil_pelatihan Mdl muZ coeff sigmaZ

```

Hasil uji citra ikan pada Klasifikasi Kesegaran Ikan Berdasarkan Ekstraksi Fitur Menggunakan Metode K-Nearest Neighbor dan Hue Saturation Value sebagai berikut:

No	Nama Citra	Klasifikasi	Hasil Klasifikasi	Keterangan
1	Segar-uji1.jpg	Ikan Segar	Ikan Segar	Benar
2	Segar-uji2.jpg	Ikan Segar	Ikan Segar	Benar
3	Segar-uji3.jpg	Ikan Segar	Ikan Segar	Benar
4	Segar-uji4.jpg	Ikan Segar	Ikan Segar	Benar
5	Segar-uji5.jpg	Ikan Segar	Ikan Segar	Benar
6	Segar-uji6.jpg	Ikan Segar	Ikan Segar	Benar
7	Segar-uji7.jpg	Ikan Segar	Ikan Segar	Benar
8	Tidaksegar-uji1.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar
9	Tidaksegar-uji2.jpg	Ikan Tidak Segar	Ikan Segar	Salah
10	Tidaksegar-uji3.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar
11	Tidaksegar-uji4.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar
12	Tidaksegar-uji5.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar
13	Tidaksegar-uji6.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar
14	Tidaksegar-uji7.jpg	Ikan Tidak Segar	Ikan Tidak Segar	Benar

KARTU BIMBINGAN SKRIPSI

Semester Gasal/Genap Tahun Akademik 2019/2020

Nama : Rizki Nursapti	Pembimbing I : Dr. Mhd. Furqan, S.Si., M. Comp. Sc
NIM : 0701163095	Pembimbing II : Abdul Halim Husugan, M. Kom.
Prog. Studi : Ilmu Komputer	SK Pembimbing :
Judul Skripsi : KLASIFIKASI PESIGARAN IKAN BERDASARKAN EKSTRAKSI FITUR MENGGUNAKAN METODE K-NEAREST NEIGHBOR (KNN) DAN Hue SATURATION VALUE (HSV)	

P E R T	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I	10/10 2020	Bab I latar Belakang		22/10 2020	BAB I (latar Belakang)	
II	15/11 2020	Bab II Tinjauan pustaka		10/11 2020	BAB II (Tinjauan Pustaka)	
III	25/11 2020	Bab III Teknik Pengumpulan Data		23/11 2020	BAB III (Teknik Pengumpulan Data)	
IV	5/12 2020	Rengsekan Proposal skripsi		2/12 2020	Rengsekan Proposal skripsi Ace proposal skripsi	
V	17/06 2021	Pengajuan Bab IV		14/06 2021	Pengajuan Bab IV	

VI	30/06 2021	Revisi Bab IV Representasi Data		25/06 2021	Revisi pada ukuran gambar di Bab IV	
VII	17/07 2021	Revisi Bab IV pada Perhitungan		8/07 2021	Revisi pada tabel 4.11 Bab IV	
VIII	22/07 2021	Pengecekan Bab IV dan Acc		20/07 2021	Acc Bab IV	
IX	12/08 2021	Pengajuan Bab V dan revisi daftar pustaka		06/08 2021	Pengajuan Bab V dan revisi	
X	25/08 2021	Acc keseluruhan		24/08 2021	Pengecekan Bab V dan Acc lengkap sbelumnya.	

Medan, 20.....
An. Dekan
Ketua Jurusan/Program Studi
ILMU KOMPUTER

ILKHA SYAFIRA, M. KOM
NIP. 198506042015031006

Catatan: Pada saat bimbingan, kartu ini harus diisi dan ditandatangani oleh pembimbing

(CURRICULUME VITAE)



Nama : Rizki Nursafitri Situmorang
Nim : 0701163095
Tempat Tanggal Lahir : Sibolga, 25 Agustus 1996
Jenis Kelamin : Perempuan
Alamat : Lopian
Kelurahan : Lopian
Kecamatan : Badiri
Kabupaten : Tapanuli Tengah
Agama : Islam
No. HP : 085261334206
Nama Orangtua
Ayah : Sahat Situmorang
Ibu : Rodia Harahap
Alamat : Lopian

Pendidikan Formal
-SD/ MI : MIN 6 Tapanuli Tengah
-SMP/ MTS : MTSN 2 Tapanuli Tengah
-SLTA/MA : SMKN 1 Sibolga