



Tesis

## Klasifikasi Ikan Menggunakan ORB-PCA dan KNN

Mirza Ramadhani  
NRP. 5116201004

Dosen Pembimbing  
Dr.Eng. Darlis Heru Murti, S.Kom., M.Kom.  
NIP. 19771217 200312 1 001

PROGRAM MAGISTER  
DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2018

*(halaman ini sengaja dikosongkan)*

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
Magister Komputer (M.Kom.)  
di  
Institut Teknologi Sepuluh Nopember Surabaya

oleh:  
Mirza Ramadhani  
Nrp. 5116201004

Dengan judul :  
Klasifikasi Ikan Menggunakan ORB-PCA dan KNN

Tanggal Ujian : 11-1-2018  
Periode Wisuda : 2017 Gasal

Disetujui oleh :

Dr. Eng. Darlis Heru Murti, S.Kom, M.Kom  
NIP. 197712172003121001



(Pembimbing 1)



Prof. Ir. Handayani Tjandrasa, M.Sc, Ph.D  
NIP. 194908231976032001

(Penguji 1)

Dr. Eng. Chastine Fatichah, S.Kom, M.Kom  
NIP. 197512202001122002



(Penguji 2)

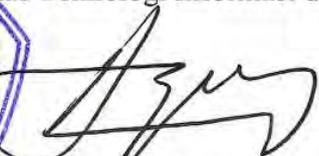
Dr. Eng. Nanik Suciati, S.Kom, M.Kom  
NIP. 197104281994122001



(Penguji 3)



Dekan, Fakultas Teknologi Informasi dan Komunikasi,



Dr. Agus Zainal Arifin, S.Kom, M.Kom.  
NIP. 19720809 199512 1 001

*(halaman ini sengaja dikosongkan)*

## **Klasifikasi Ikan Menggunakan ORB-PCA dan KNN**

Nama mahasiswa : Mirza Ramadhani  
NRP : 5116201004  
Pembimbing : Dr. Eng. Darlis Herumurti, S.Kom, M.Kom

### **ABSTRAK**

Ikan memiliki bentuk dan ukuran tertentu yang berbeda antara ikan yang satu dengan yang lain. Permasalahan dalam mengenali jenis ikan lebih kompleks dibandingkan dengan mengenali wajah manusia. Perbedaan bentuk, warna, dan tekstur pada ikan lebih bervariasi dibandingkan manusia. Pengenalan jenis ikan pada umumnya masih dilakukan secara manual menggunakan pengamatan mata. Sehingga diperlukan adanya sistem yang dapat mengenali ikan secara otomatis. Penelitian sebelumnya juga sudah dapat mengenali jenis ikan namun sensitive terhadap berbagai transformasi atau deformasi dari sebuah objek, dan waktu komputasi yang tidak sedikit, sehingga kurang efektif untuk mengenali objek ikan.

Dalam Penelitian ini, kami mengusulkan metode untuk mendeteksi dan mengenali jenis objek ikan menggunakan metode ORB-PCA dan KNN. Pengaplikasian dari metode ORB diterapkan untuk ekstraksi fitur dari gambar yang diambil. PCA digunakan untuk mereduksi dimensionalitas dari citra. Kemudian hasil tersebut akan diklasifikasi menggunakan KNN untuk menentukan label kelas yang tepat dari input data ikan. Hasil uji coba menunjukkan bahwa metode yang diusulkan pada penelitian ini mencapai akurasi klasifikasi sebesar 97.5%.

**Kata kunci:** deteksi ikan, pengenalan ikan, klasifikasi, ORB, PCA, KNN

*(halaman ini sengaja dikosongkan)*

## **Fish Classification Using ORB-PCA and KNN**

Name : Mirza Ramadhani  
Student ID : 5116201004  
Supervisor : Dr. Eng. Darlis Herumurti, S.Kom, M.Kom

### **ABSTRACT**

Fish have a certain shape and size that differ between fish one with another. The problem of recognizing fish species is more complex than recognizing human faces. Differences in shapes, colors, and textures in fish are more varied than humans. The fish recognition of species in general is still done manually using eye observation. So that required a system that can recognize fish automatically. Previous research has also been able to recognize fish species but is sensitive to various transformations or deformations of an object, and computational time is not small, making it less effective to recognize fish.

In this study, we propose methods for detecting and recognizing fish object types using the ORB-PCA and KNN methods. The application of the ORB method is applied to feature extraction from captured images. PCA is used to reduce dimensionality of the image. Then the results will be classified using KNN to determine the exact class label of the fish data input. The experimental results showed that the proposed method reached classification accuracy by 97.5%.

**Keywords:** *fish detection, fish recognition, classification, ORB, PCA, KNN*

*(halaman ini sengaja dikosongkan)*

## KATA PENGANTAR

*Alhamdulillahirabbil'alamin.* Puji dan syukur penulis panjatkan kehadirat Allah SWT atas berkat, rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan buku tesis yang berjudul “Klasifikasi Ikan Menggunakan ORB-PCA dan KNN” ini, dan tak lupa pula salam dan salawat kepada junjungan kita Rasulullah SAW. Tesis ini dibuat sebagai salah satu syarat dalam menyelesaikan Program Studi Magister di Institut Teknologi Sepuluh Nopember Surabaya.

Pada kesempatan ini, dengan kerendahan hati penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan kasih sayang, dukungan, bantuan, motivasi, nasehat, doa serta bimbingannya baik secara moril maupun material yang telah diberikan kepada saya, khususnya kepada:

1. Seluruh keluarga, khususnya Ibu dan bapak tercinta yang tiada henti selalu mendukung saya, selalu mendoakan anaknya yang terbaik dan selalu menjadi panutan yang baik.
2. Bapak Dr. Eng. Darlis Herumurti, S.Kom, M.Kom selaku pembimbing yang telah banyak meluangkan waktu dan pikiran untuk senantiasa selalu memberikan arahan dan bimbingan kepada penulis. Semoga Allah SWT senantiasa merahmati bapak dan keluarga.
3. Ibu Prof.Ir.Handayani Tjandrasa, M.Sc, Ph.D., Dr. Eng. Chastine Fatichah, S.Kom, M.Kom, dan Dr. Eng. Nanik Suciati, S.Kom, M.Kom selaku dosen penguji yang telah memberikan banyak kritik, saran dan masukan demi kesempurnaan tesis ini.
4. Seluruh dosen Pascasarjana Teknik Informatika yang telah memberikan ilmu dan pengetahuan kepada penulis selama menempuh studi.
5. Pasangan hidup saya, Almira Faliyanti, yang selalu memberikan perhatian, kasih sayang, motivasi dan semangat dalam menyelesaikan studi dan Tesis ini. Semoga Allah SWT selalu melancarkan urusan dan segala rencana kita untuk kedepannya.
6. Rekan-rekan seperjuangan, Ihsan Alfani dan Wanvy Arifha Saputra, yang telah banyak membantu saya selama menempuh studi dengan memberikan berbagai

informasi, membagikan pengalamannya dalam menempuh studi. Semoga Allah SWT akan membalas segala kebaikan hati kalian.

7. Para Sahabat serta teman lainnya yang tidak dapat disebutkan satu persatu, dan semua pihak yang telah banyak membantu proses penyelesaian tesis ini. terima kasih atas bantuan dan motivasi yang telah diberikan.

Akhirnya dengan segala kerendahan hati penulis menyadari masih banyak terdapat kekurangan pada Tesis ini. Oleh karena itu, segala tegur sapa dan kritik yang sifatnya membangun sangat penulis harapkan demi kesempurnaan Tesis ini. Penulis berharap bahwa perbuatan baik dari semua orang yang dengan tulus memberikan kontribusi terhadap penyusunan Tesis ini mendapatkan pahala dari Allah. Aamiin Alluhamma Aamiin.

Surabaya, Januari 2018

Penulis

# DAFTAR ISI

Halaman

LEMBAR PENGESAHAN .....	iii
ABSTRAK .....	v
ABSTRACT .....	vii
KATA PENGANTAR.....	ix
DAFTAR ISI .....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL .....	xv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah .....	6
1.3 Tujuan .....	6
1.4 Manfaat .....	6
1.5 Batasan Masalah .....	6
1.6 Kontribusi .....	6
BAB II LANDASAN TEORI .....	9
2.1 Identifikasi dan klasifikasi ikan .....	9
2.2 <i>Computer Vision</i> .....	11
2.3 <i>Tracking</i> dan Pengenalan Objek.....	13
2.4 OpenCV .....	15
2.5 Feature/Keypoint Detector.....	16
2.6 ORB ( <i>Oriented FAST and Rotated BRIEF</i> ).....	16
2.7 PCA ( <i>Principal Component Analysis</i> ).....	19
2.8 K-Nearest Neighbor (KNN).....	21
BAB III METODE PENELITIAN .....	23
3.1 Studi Literatur .....	23
3.2 Pengumpulan Data.....	23
3.3 Metode Usulan.....	24
3.3.1 <i>ORB (Oriented FAST Rotated BRIEF) Detector</i> .....	26
3.3.2 <i>PCA (Principal Component Analysis)</i> .....	29
3.3.3 Klasifikasi Menggunakan <i>K-Nearest Neighbor (KNN)</i> .....	30

3.4	Analisis Hasil .....	31
BAB 4 UJI COBA DAN ANALISA HASIL .....		33
4.1	Lingkungan Uji Coba.....	33
4.2	Data Uji Coba .....	33
4.3	Skenario Uji Coba.....	34
4.3.1	Pengujian Ekstraksi Fitur .....	35
4.3.2	Pengujian Metode Yang Diusulkan.....	37
4.3.3	Pengujian Dengan Metode Lain.....	48
4.4	Analisa Hasil Uji Coba .....	50
BAB 5 KESIMPULAN DAN SARAN .....		54
5.1	Kesimpulan .....	54
5.2	Saran .....	54
DAFTAR PUSTAKA.....		56
LAMPIRAN A .....		60
LAMPIRAN B.....		61
BIODATA PENULIS.....		62

## DAFTAR GAMBAR

	Halaman
<b>Gambar 2.1</b> Skema ikan .....	10
<b>Gambar 2.2</b> Struktur Computer Vision Secara Umum .....	12
<b>Gambar 2.3</b> Contoh Penerapan Feature Matching .....	14
<b>Gambar 2.4</b> Penerapan Feature Tracking .....	14
<b>Gambar 2.5</b> Contoh Hasil Pencocokan Menggunakan ORB .....	17
<b>Gambar 3.1</b> Tahapan Penelitian .....	23
<b>Gambar 3.2</b> <i>Sample</i> Data Pengujian .....	24
<b>Gambar 3.3</b> Metode yang diusulkan .....	24
<b>Gambar 3.4</b> Gambaran detail proses metode yang diusulkan .....	25
<b>Gambar 3.5</b> Hasil deteksi keypoint menggunakan ORB .....	29
<b>Gambar 3.6</b> Hasil klasifikasi ikan menggunakan KNN .....	31
<b>Gambar 4.1</b> Data <i>testing</i> .....	33
<b>Gambar 4.2</b> Data <i>training</i> .....	34
<b>Gambar 4.3</b> Citra input .....	35
<b>Gambar 4.4</b> Keypoint yang terdeteksi menggunakan ORB .....	36
<b>Gambar 4.5</b> Matriks hasil deskripsi fitur ORB .....	37
<b>Gambar 4.6</b> Citra input data pengujian penentuan nilai $k$ .....	39
<b>Gambar 4.7</b> Perbandingan tingkat akurasi setiap nilai $k$ .....	40
<b>Gambar 4.8</b> (a) input (b) output hasil 10 rank terbaik untuk penentuan label kelas .....	41
<b>Gambar 4.9</b> Label kelas hasil klasifikasi dari satu jenis ikan .....	43
<b>Gambar 4.10</b> Input pengujian rotasi .....	44
<b>Gambar 4.11</b> Hasil 10 rank ketetanggan terdekat pengujian rotasi .....	44
<b>Gambar 4.12</b> Hasil pengujian pengenalan ikan dengan perbedaan rotasi .....	45
<b>Gambar 4.13</b> Perbandingan akurasi metode klasifikasi ikan .....	51
<b>Gambar 4.14</b> Perbandingan waktu komputasi metode klasifikasi ikan .....	52

*(halaman ini sengaja dikosongkan)*

## DAFTAR TABEL

	Halaman
Tabel 4.1 Hasil komparatif penentuan nilai K.....	39
Tabel 4.2 Rincian Hasil Klasifikasi Citra Ikan.....	41
Tabel 4.3 Hasil Pengujian klasifikasi dengan berbagai sudut rotasi .....	45
Tabel 4.4 Hasil Klasifikasi Keseluruhan Data.....	46
Tabel 4.5 Waktu komputasi proses klasifikasi keseluruhan data .....	47
Tabel 4.6 Hasil klasifikasi keseluruhan data menggunakan metode ORB-KNN.....	48
Tabel 4.7 Hasil klasifikasi menggunakan metode BRISK-PCA-KNN .....	49
Tabel 4.8 Hasil klasifikasi menggunakan metode BRISK-KNN .....	50

*(halaman ini sengaja dikosongkan)*

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Ikan adalah hewan berdarah dingin, ciri khasnya adalah mempunyai tulang belakang, insang dan sirip, dan terutama ikan sangat bergantung atas air sebagai medium dimana tempat mereka tinggal. Ikan memiliki kemampuan di dalam air untuk bergerak dengan menggunakan sirip untuk menjaga keseimbangan tubuhnya sehingga tidak tergantung pada arus atau gerakan air yang disebabkan oleh arah angin. Dalam keluarga hewan bertulang belakang/ vertebrata, ikan menempati jumlah terbesar, sampai sekarang terdapat sekitar 25.000 species yang tercatat, walaupun perkiraannya ada pada kisaran 40.000 spesies, yang terdiri dari 483 famili dan 57 ordo. Jenis-jenis ikan ini sebagian besar tersebar di perairan laut yaitu sekitar 58% (13,630 jenis) dan 42% (9870 jenis) dari keseluruhan jenis ikan. Jumlah jenis ikan yang lebih besar berada di perairan laut, dapat dipahami bahwa hampir 70% permukaan bumi ini terdiri dari air laut dan hanya sekitar 1% merupakan perairan tawar. Mereka bukan hanya dibedakan oleh jumlah spesies yang beragam, tetapi juga berbeda dalam berbagai ukuran dan bentuk. Mulai dari ikan yang berukuran kecil yang disebut Percid dari Amerika (*Etheostoma microperca*) yang dewasa secara seksual pada ukuran 27 mm. Di samping itu ada juga jenis goby dari Pacifik (*Eviota* sp.) yang bertelur pada ukuran kurang dari 15 mm. Ada pula yang berukuran raksasa seperti Hiu (*Rhincodon* sp.) yang dapat mencapai panjang 21 meter dengan berat 25 ton lebih. Kebanyakan dari ikan memiliki bentuk seperti terpedo, walaupun beberapa diantaranya ada yang berbentuk flat dan terdapat bentuk-bentuk yang lainnya yang lain dari yang lain. (Burhanuddin, 2008).

Keanekaragaman tempat hidup mempengaruhi ikan penghuninya. Banyak variasi yang tak terhitung jumlahnya pada ikan yang menyangkut masalah struktur, bentuk, sirip dan sebagainya, merupakan modifikasi yang dikembangkan ikan dalam usahanya untuk menyesuaikan diri terhadap suatu lingkungan tertentu. Sungai yang deras dan sungai yang tenang memiliki arus yang berbeda sehingga mempengaruhi kehidupan ikan. Danau yang dangkal dan yang dalam mempunyai berbagai pola perubahan suhu secara musiman. Kedalaman samudra menyajikan kemungkinan untuk

pegkhususan yang lain. Lingkungan perairan samudra yang tampak sama di berbagai daerah di dunia ini sebetulnya sama sekali berbeda dalam hal sifat kimiawi airnya, tipe dasarnya dan perubahan musimnya. Ikan menyesuaikan diri terhadap segala kondisi tersebut.

Dari keanekaragaman tempat hidup tersebut, Ikan memiliki bentuk dan ukuran tertentu dan berbeda antara ikan yang satu dengan yang lain. Hal ini menunjukkan bahwa ada spesifikasi tertentu pada karakteristik, bentuk dan ukuran tubuh ikan di alam. Sehingga dari banyaknya keberagaman dan perbedaan karakteristik dari ikan, sehingga sangat sulit bagi kita untuk mengenali keberagaman ikan tersebut jika tidak mempunyai ilmu tentang perikanan. Dalam perusahaan produksi ikan, selama ini pengenalan jenis ikan pun masih dilakukan secara manual melalui pengamatan mata. Klasifikasi secara manual tersebut akan membutuhkan waktu yang lama, apalagi dalam jumlah yang sangat banyak. Sehingga diperlukan adanya sistem yang dapat mengenali dan mengklasifikasi keberagaman ikan secara otomatis.

Dalam mempelajari proses pengenalan ikan. Maka perlu adanya pengetahuan tentang bagaimana proses klasifikasi objek. Pada penelitian sebelumnya, sudah banyak peneliti yang membuat sistem tentang pengenalan objek, namun yang paling populer yaitu *face recognition*. *Face recognition* yang pernah dilakukan oleh (Mulyono, 2012) menjelaskan tentang penggunaan *eigenface* berbasis PCA (Principal Components Analysis) untuk mengenali emosi wajah. Eigenfaces membentuk basis set dari semua gambar yang digunakan untuk membangun matriks kovarians. Ini menghasilkan reduksi dimensi dengan memungkinkan set yang lebih kecil dari dasar gambar untuk mewakili gambar pelatihan asli. Klasifikasi dapat dicapai dengan membandingkan bagaimana wajah diwakili oleh basis set. Namun metode ini sangat sensitif dalam mengenali objek dengan pencahayaan tidak merata, sehingga tidak efektif untuk digunakan dalam mengenali jenis ikan. Kemudian penelitian sebelumnya yang dilakukan oleh (Jiang, Liu, Yang, & Liao, 2013), Algoritma Fisherface menggunakan pendekatan 2 buah metode, yaitu PCA (*Principal Component Analysis*) and FLD (Fisher Linear Discriminant). Setelah gambar dipetakan ke dalam ruang eigen menggunakan PCA (*Principal Component Analysis*), hasil perhitungan tersebut akan dikonversi ke dalam subruang linier menggunakan FLD (Fisher Linear Discriminant). Teknik PCA (*Principal*

*Component Analysis*) ini juga digunakan pada perhitungan Algoritma Eigenface. Pada penelitian tersebut membuktikan bahwa PCA sangat akurat dalam mengenali wajah.

Alur kerja sistem *face recognition* dapat dijadikan acuan dalam penelitian ini, karena cara kerja *face recognition* memang sangat mirip dengan alur kerja sistem untuk mengenali jenis ikan. Namun cara kerja pengenalan ikan lebih kompleks dibandingkan dengan pengenalan wajah. Dari berbagai macam ras manusia, manusia memiliki perbedaan jenis warna kulit dan banyak penggolongan yang biasanya didasarkan pada ras dan keturunan. Warna kulit manusia mempunyai warna tingkat kecoklatan yang berbeda. Sehingga, untuk mendeteksi dan mengenali wajah manusia, sistem akan mendeteksi dari warna kulit terlebih dahulu untuk mendeteksi apakah itu wajah manusia atau tidak. Berbeda halnya dengan ikan, ikan dapat dikategorikan dengan ikan air tawar dan air laut yang tersebar diseluruh lautan dunia. Laut dunia memiliki puluhan hingga ratusan ribu jenis spesies ikan. Dari banyaknya spesies ikan tersebut, ikan memiliki warna atau pola yang lebih bervariasi dibandingkan manusia. Tidak hanya dari warna saja, ikan juga memiliki bentuk dan ukuran yang lebih bervariasi dibandingkan manusia. Manusia mempunyai *gesture* wajah yang berbeda-beda, dari bentuk kepala, mata, telinga, hidung, atau mulut. Namun pada dasarnya, anatomi wajah manusia tetap sama. Berbeda dengan ikan, setiap spesiesnya mempunyai bentuk lebih bervariasi yang dibedakan oleh bentuk tubuh, bentuk kepala, jumlah dan letak sirip, dan bentuk ekor. Sehingga permasalahan sistem dalam mengenali jenis ikan lebih kompleks dibandingkan mengenali wajah manusia.

Beberapa penelitian sebelumnya juga sudah ada yang meneliti tentang klasifikasi ikan. Salah satunya penelitian dengan judul “Ekstraksi Fitur Berdasarkan Deskriptor Bentuk dan Titik Salien untuk Klasifikasi Citra Ikan Tuna” yang dilakukan oleh (R.A. Pawening,2016). Penelitian tersebut mengembangkan sistem ekstraksi fitur baru yang mengintegrasikan deskriptor bentuk dan titik salien untuk klasifikasi citra ikan tuna. Citra input dilakukan segmentasi untuk memisahkan objek dengan background dengan mengambil kanal Hue pada citra HSV hasil transformasi. Selanjutnya citra dilakukan proses morfologi untuk menghilangkan noise sehingga memperoleh hasil segmentasi terbaik. Citra objek yang sudah terpisah adalah berupa citra biner dan citra grayscale untuk proses ekstraksi fitur. Pada citra biner, deskriptor

bentuk diekstrak menggunakan Fourier Descriptor (FD) dan pada citra grayscale, titik salien diekstrak menggunakan Speeded Up Robust Features (SURF). Proses klasifikasi memerlukan dimensi fitur yang sama. Untuk mendapatkan fitur titik salien dengan dimensi yang sama dilakukan pemrosesan menggunakan Bag of Feature (BOF). Kedua jenis fitur yang sudah diperoleh dilakukan integrasi dengan mempertimbangkan bobot masing-masing fitur. Uji coba dilakukan pada dataset 3 jenis ikan tuna dengan 10-fold cross validation. Hasil uji coba menunjukkan bahwa dengan mengintegrasikan deskriptor bentuk dan titik salien permasalahan klasifikasi ikan tuna dengan bentuk yang mirip dapat diselesaikan. Namun penelitian tersebut sangat bergantung kepada hasil segmentasi, dan hanya dapat mengklasifikasi objek ikan dengan background yang sama di semua datasetnya.

Ekstraksi fitur merupakan salah satu tahapan penting dalam proses klasifikasi. Sehingga diperlukan pemilihan metode yang tepat dalam mengekstraksi fitur agar proses klasifikasi lebih akurat. Seperti pada Penelitian oleh Ethan Rublee dengan judul “*ORB: an efficient alternative to SIFT or SURF*”. Penelitian ini melakukan penelitian tentang *feature matching* antara beberapa metode ekstraksi fitur seperti SURF (*Speeded-Up Robust Features*), SIFT (*Scale-Invariant Feature Transform*), dan ORB (*Oriented FAST and Rotated BRIEF*). Penelitian tersebut melakukan pengujian terhadap keypoint dan deskriptor yang dihasilkan terhadap masing-masing metode. Uji coba yang dilakukan menunjukkan bahwa proses ORB lebih cepat dalam mendeteksi keypoint dari objek dibandingkan SIFT dan SURF diberbagai situasi, karena ORB memiliki komputasi yang rendah. Kemudian metode ORB juga tahan terhadap pencahayaan, blur, dan noise. Metode ini diujicoba menggunakan aplikasi yang menggunakan deteksi objek dan *patch tracking* pada *smartphone* (Rublee, 2011). Penelitian tersebut membuktikan bahwa ekstraksi fitur menggunakan ORB sangat handal dalam berbagai situasi seperti pencahayaan, blur, dan noise. Metode ORB juga mempunyai komputasi yang rendah dan tahan terhadap transformasi dan deformasi, misalnya perubahan rotasi atau pergerakan dari objek. Sehingga metode ini sangat cocok digunakan mengekstraksi fitur dalam klasifikasi.

Penelitian dengan metode ekstraksi fitur terbaru, yang dilakukan oleh (Leutenegger, 2011) berjudul “BRISK : Binary Robust Invariant Scalable Keypoints”,

seperti layaknya ORB, BRISK adalah sebuah metode baru untuk deteksi, deskripsi dan deteksi keypoint. Sebuah evaluasi komprehensif terhadap dataset benchmark menunjukkan kinerja BRISK yang adaptif dan berkualitas tinggi seperti algoritma mutakhir, walaupun dengan biaya komputasi yang jauh lebih rendah (urutan besarnya lebih cepat dari pada SURF dalam beberapa kasus). Kunci untuk mempercepat terletak pada penerapan ruang skala baru berbasis FAST detektor yang dikombinasikan dengan perakitan deskriptor bit-string dari perbandingan intensitas yang diambil dengan sampling khusus dari setiap lingkungan pengetesan. Metode ini telah diuji coba dengan dibandingkan dengan metode lainnya, seperti SIFT dan SURF. Hasil menunjukkan bahwa kinerja BRISK dapat mendeteksi keypoint yang lebih akurat dan invarian terhadap rotasi. Metode ini akan dijadikan metode pembanding dalam penelitian ini untuk mengetahui tingkat akurasi dan waktu komputasi yang dibutuhkan untuk melakukan klasifikasi ikan.

Penelitian dengan judul “Pengenalan Jenis-jenis Ikan Menggunakan Metode Analisis Komponen Utama” yang dilakukan oleh (Santoso,2006). Metode yang digunakan untuk klasifikasi ikan adalah menggunakan PCA (*Principal Component Analysis*) yang digunakan untuk mendeteksi jenis ikan, kemudian menggunakan KNN (*K-Nearest Neighbor*) untuk proses mengklasifikasi ikan. Hasil dari metode KNN membuat tingkat keberhasilan pengenalan pada pengujian cukup tinggi. Metode tersebut juga mampu mengurangi kesalahan pengenalan yang terjadi pada pengujian dengan jumlah citra latih yang lebih sedikit. Namun hasil klasifikasi sangat ditentukan oleh kualitas gambar, seperti faktor cahaya, noise, blur, dan lain-lain. Karena untuk dapat mengklasifikasi objek dengan lebih akurat, diperlukan metode yang invariant terhadap segala perubahan. Namun dari penelitian tersebut, klasifikasi menggunakan KNN dapat menjadi pilihan yang tepat, karena terbukti dapat mengklasifikasi ikan dengan tingkat akurasi yang tinggi.

Dari berbagai permasalahan dalam penelitian tentang klasifikasi ikan tersebut, sehingga diperlukan macam perubahan seperti perbedaan ukuran atau perubahan rotasi pada citra. KNN juga dapat dijadikan s semakin besar, sehingga semakin besar pula kemungkinan bahwa *Nearest Neighbour* dari suatu instance sebetulnya sama sekali tidak “near“. Masalah tingginya dimensionalitas bisa diatasi dengan pemilihan variabel atau rekayasa fitur dengan PCA.

Sehingga dalam penelitian ini, metode yang digunakan yaitu menggunakan ORB-PCA dan KNN.

## **1.2 Perumusan Masalah**

Dari latar belakang diatas didapatkan permasalahan sebagai berikut :

1. Bagaimana ekstraksi fitur menggunakan ORB?
2. Bagaimana mereduksi dimensionalitas menggunakan PCA?
3. Bagaimana mengklasifikasi ikan menggunakan KNN?

## **1.3 Tujuan**

Penelitian ini bertujuan untuk mengklasifikasi berbagai jenis ikan dengan otomatis. Serta menunjukk

## **1.4 Manfaat**

Manfaat dari penelitian ini adalah sebagai media pengenalan dan pembelajaran tentang jenis ikan. K

## **1.5 Batasan Masalah**

Melihat luasnya permasalahan, maka dalam penelitian ini dibuat beberapa asumsi dengan tujuan agar pembahasan menjadi lebih terarah serta untuk menyederhanakan dan membatasi permasalahan. Adapun batasan masalah dari penelitian ini adalah sebagai berikut:

1. Pembuatan proses klasifikasi ikan menggunakan metode ORB, PCA, KNN, dan *library* OpenCV.
2. Citra ikan yang dikenali tidak memerlukan proses *pre-processing* karena karena sebagian citra telah disegmentasi secara manual dan akan dilakukan pengujian dengan gambar latar belakang yang berbeda.
3. Aplikasi yang dirancang dapat mengenali 15 jenis citra ikan yang berbeda.

## **1.6 Kontribusi**

Kontribusi dari penelitian ini adalah untuk dapat mengenali ikan yang mempunyai keberagaman bentuk disetiap jenisnya menggunakan metode *Oriented FAST and*

*Rotated BRIEF* dan *Principal Component Analysis*, kemudian diklasifikasi menggunakan metode *K-Nearest Neighbor*.

*(halaman ini sengaja dikosongkan)*

## **BAB II**

### **LANDASAN TEORI**

Bab ini akan membahas secara singkat mengenai landasan teori yang digunakan pada penelitian ini. Teori mengenai klasifikasi ikan yang menjelaskan tentang berbagai macam karakteristik dari ikan yang akan diidentifikasi untuk mengenali ikan disajikan dalam Bab ini. Teori tentang computer vision, maupun teori tentang pengenalan objek juga akan dijelaskan secara mendasar pada bab ini. Dalam Bab ini, juga akan menjelaskan teori mengenai berbagai metode yang digunakan pada penelitian.

#### **2.1 Identifikasi dan klasifikasi ikan**

Mayr dalam Layli (2006) mengatakan bahwa ikan sebagai salah satu organisme yang menjadi kajian ekologi, sehingga harus dijaga kelestariannya. Sebagai langkah awal diperlukan kegiatan identifikasi terhadap organisme tersebut. Identifikasi adalah menempatkan atau memberikan identitas suatu individu melalui prosedur deduktif ke dalam suatu takson dengan menggunakan kunci determinasi.

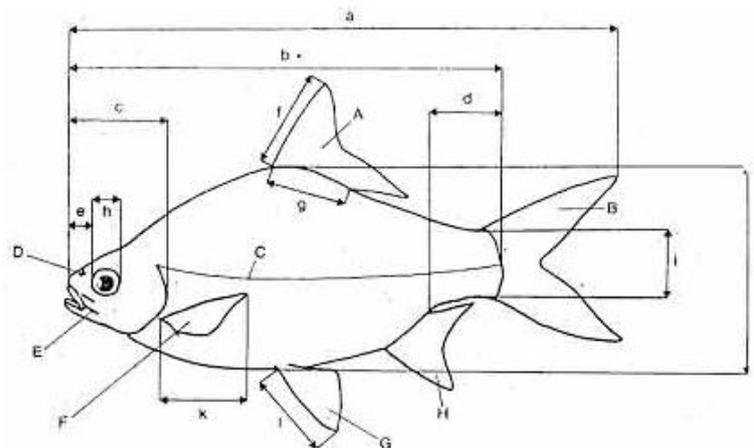
Kunci determinasi adalah kunci jawaban yang digunakan untuk menetapkan identitas suatu individu. Kegiatan identifikasi bertujuan untuk mencari dan mengenal ciri-ciri taksonomi yang sangat bervariasi dan memasukkannya ke dalam suatu takson. Selain itu untuk mengetahui nama suatu individu atau spesies dengan cara mengamati beberapa karakter atau ciri morfologi spesies tersebut dengan membandingkan ciri-ciri yang ada sesuai dengan kunci determinasi.

Ikan dibedakan berdasarkan karakter-karakter umum yang dapat membedakan antara kelompok yang satu dengan kelompok yang lain. Adapun karakter-karakter yang biasa digunakan dalam identifikasi ikan antara lain, yaitu: bentuk umum tubuh, bentuk dan jumlah sirip, bentuk mulut, bentuk ekor, dan perbandingan dan posisi anggota tubuh (Adrim, 2010).

Klasifikasi ialah menetapkan definisi dari kelompok atau kategori menurut skala hierarki. Tiap-tiap kategori ini meliputi satu atau beberapa kelompok rendah yang terdekat, yang merupakan kategori lebih rendah berikutnya (Saenin, 1968).

Pengenalan struktur ikan tidak terlepas dari morfologi ikan yaitu bentuk luar ikan yang merupakan ciri-ciri yang mudah dilihat dan diingat dalam mempelajari jenis-jenis ikan. Morfologi ikan sangat berhubungan dengan habitat ikan tersebut di perairan. Bagian-bagian tubuh ikan akan digunakan dalam proses identifikasi.

Ikan memiliki ukuran yang berbeda-beda di setiap spesiesnya. Perbedaan ukuran tersebut mencakup dari berbagai organ pada ikan. Ukuran standar yang dipakai dapat dilihat pada Gambar 2.1. Pada gambar tersebut, ada beberapa bagian yang dapat digunakan dalam identifikasi jenis ikan.



**Gambar 2.1** Skema ikan untuk menunjukkan bagian-bagian utama ikan dan ukuran-ukuran yang digunakan dalam identifikasi. (A) sirip punggung, (B) sirip ekor, (C) gurat sisi, (D) lubang hidung, (E) sungut, (F) sirip dada, (G) sirip perut, (H) sirip dubur, (a) panjang total, (b) panjang standar, (c) panjang kepala, (d) panjang batang ekor, (e) panjang moncong, (f) tinggi sirip punggung, (g) panjang pangkal sirip punggung, (h) diameter mata, (i) tinggi batang ekor, (j) tinggi badan, (k) panjang sirip dada, (l) panjang sirip perut. (Gonzalez & Woodz, 2007)

Semua ukuran yang digunakan merupakan pengukuran yang diambil dari satu titik ke titik lain tanpa melalui lengkungan badan.

- Panjang total (TL) diukur mulai dari bagian terdepan moncong/bibir (premaxillae) hingga ujung ekor.

- Panjang standar (SL) diukur mulai dari bagian terdepan moncong/bibir (premaxillae) hingga pertengahan pangkal sirip ekor (pangkal sirip ekor bukan berarti sisik terakhir karena sisik-sisik tersebut biasanya memanjang sampai ke sirip ekor)
- Panjang kepala (HL) diukur mulai dari bagian terdepan moncong/bibir (premaxilla) hingga bagian terbelakang operculum atau membran operculum.
- Panjang batang ekor (LCP) diukur mulai dari jari terakhir sirip dubur hingga pertengahan pangkal batang ekor.
- Panjang moncong (SNL) diukur mulai dari bagian terdepan moncong/bibir hingga pertengahan garis vertikal yang menghubungkan bagian anterior mata.
- Tinggi sirip punggung (DD) diukur mulai dari pangkal hingga ujung pada jari-jari pertama sirip punggung.
- Diameter mata (ED) diukur mulai dari bagian anterior hingga posterior bola mata, diukur mengikuti garis horisontal.
- Tinggi batang ekor (DCP) diukur mulai dari bagian dorsal hingga ventral pangkal ekor.
- Tinggi badan diukur (BD) secara vertikal mulai dari pangkal jari-jari pertama sirip punggung hingga pangkal jari-jari pertama sirip perut.
- Panjang sirip dada diukur mulai dari pangkal hingga ujung jari-jari sirip dada.
- Panjang sirip perut diukur mulai dari pangkal hingga ujung sirip perut.

## ***2.2 Computer Vision***

Computer vision adalah ilmu dan metode aplikasi dalam menggunakan komputer untuk memahami isi citra (image content). Area permasalahan dalam computer vision adalah pengukuran dan pemrosesan, yang dapat dilakukan dengan berbagai metode. Beberapa area penelitian computer vision adalah sebagai berikut:

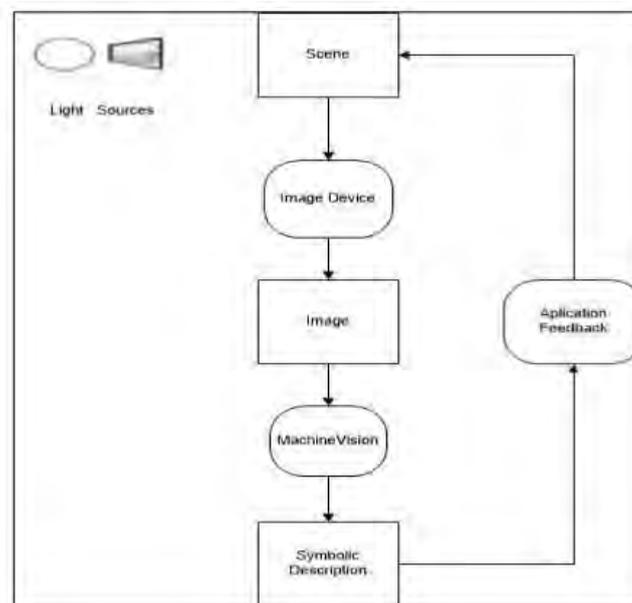
1. Recognition/Pengenalan bertujuan mengenali objek data citra, aplikasinya seperti Content Based Image Retrieval (CBIR), Optical Character Recognition (OCR).

2. Motion/gerakan bertujuan mengenali data citra bergerak. Aplikasinya seperti Egomotion yang membagi gerakan 3D dari kamera, tracking yang memperkirakan satu atau beberapa objek dalam citra.

3. Restorasi citra, bertujuan untuk mendapatkan data citra, citra bergerak atau objek 3D tanpa noise. Content Based Image Retrieval (Temu Kembali Citra Berbasis Isi) adalah salah satu aplikasi dari computer vision dalam permasalahan temu kembali citra dalam jumlah data citra yang besar. Arti "Berbasis Isi" adalah pencarian citra berdasarkan isi citra. Terminologi CBIR dikenalkan pada tahun 1992 oleh T.Kato, dalam uji coba nya dijelaskan sistem temu kembali citra dari database citra.

Pengolahan citra merupakan proses awal (preprocessing) pada computer vision, sedangkan pengenalan pola merupakan proses untuk menginterpretasi citra. Teknik-teknik di dalam pengenalan pola memainkan peranan penting dalam computer vision untuk mengenali objek (Aditya, 2013).

Elemen-elemen *Computer Vision* mempunyai struktur yang mendasari suatu mesin vision seperti pada Gambar 2.2.



**Gambar 2.2** Struktur Computer Vision Secara Umum

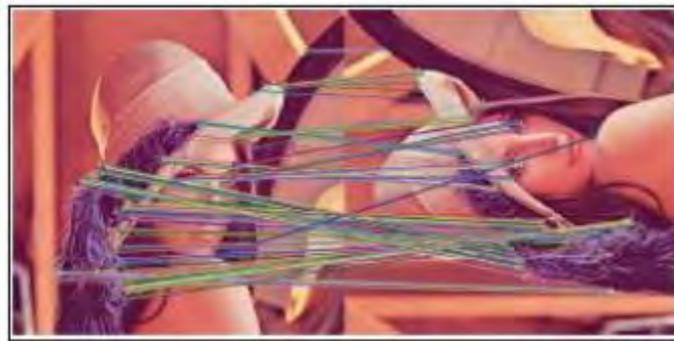
(Sumber: <http://elib.unikom.ac.id/download.php?id=90697>)

Keterangan Gambar 2.2 diatas adalah sebagai berikut:

1. *Light sources*, merupakan sumber cahaya yang digunakan sebagai sumber untuk aplikasi seperti layar laser, sistem robotika dan sebagainya.
2. *Scene*, merupakan kumpulan objek.
3. *Image Device*, merupakan alat yang digunakan untuk mengubah gambar menjadi suatu yang dimengerti oleh mesin.
4. *Image*, merupakan gambar/citra dari suatu objek yang merupakan representasi dari keadaan sesungguhnya.
5. *Machine vision*, merupakan mesin yang menginterpretasikan gambar yang berkenaan dengan ciri-ciri pola maupun objek yang dapat ditelusuri oleh sistem.
6. *Symbolic description*, merupakan sistem yang dapat digunakan untuk menganalogikan struktur kerja sistem ke simbol-simbol tertentu yang dapat dimengerti oleh sistem.
7. *Application feedback*, merupakan suatu keadaan yang dapat memberikan respon untuk menerima gambar dari suatu sistem penglihatan.

### **2.3 Tracking dan Pengenalan Objek**

*Tracking* dan pengenalan objek memiliki ruang lingkup yang luas dan sangat menjanjikan apabila diterapkan pada suatu aplikasi. Proses untuk dapat mengenali objek dan melakukan *tracking* biasanya menggunakan banyak algoritma misalnya yang populer yakni berbasis fitur. *Tracking* berbasis fitur merupakan suatu tindakan untuk menemukan sifat khas (fitur) dari suatu citra pada suatu *scene*, seperti pada Gambar 2.3 (Peter,2014).



**Gambar 2.3** Contoh Penerapan Feature Matching

(Sumber: Peter (2014). *Computer Vision in ArrayFire - Part 2: Feature Description and Matching*  
<http://arrayfire.com/computer-vision-in-arrayfire-part-2/>)

*Matching* merupakan pencocokan fitur dari 2 objek seperti pada Gambar 2.3. *Matching* antar *keypoint* dari fitur-fitur yang mempunyai kecocokan dan ditegaskan dengan garis-garis penghubung seperti pada Gambar 2.3. Pengenalan Objek (*Object Recognition*) merupakan tindakan yang digunakan untuk mengidentifikasi suatu objek berdasarkan geometri seperti pada Gambar 2.4.



**Gambar 2.4** Penerapan Feature Tracking

(Sumber: *Visual Geometry Group (2004). University of Oxford*,  
<http://www.robots.ox.ac.uk/~vgg/data/data-aff.html>)

*Feature Tracking* seperti pada Gambar 2.4 berhasil melakukan *tracking* terhadap citra yang sama dengan perbedaan geometri citra. *Tracking* ditandai dengan munculnya efek *Homography* berbentuk persegi pada citra.

*Tracking* berbasis fitur dalam penerapannya secara umum memiliki 3 tahapan yang dilakukan yakni:

1. Deteksi fitur (*Feature Detection*)

Merupakan proses untuk menemukan sifat yang sangat khas dari suatu objek salah satu contohnya yakni sudut.

2. Ekstraksi Deskriptor (*Descriptor extraction*)

Merupakan proses pengkodean *patch* tekstur yang ada pada setiap fitur.

3. *Descriptor matching*

Merupakan proses untuk menemukan tekstur *patch* yang mirip pada citra atau gambar yang berbeda.

*Tracking* berbasis fitur dalam implementasinya terdapat beberapa metode yang digunakan dan dikembangkan dalam beberapa penelitian seperti: *Scale Invariant Feature Transform* (SIFT), *Speeded Up Robust Features* (SURF), FAST, BRIEF dan ORB (Howse, 2013).

## 2.4 OpenCV

*Open Computer Vision* (OpenCV) adalah sebuah *Application Programming Interface* (API) *Library* yang sudah sangat familiar pada Pengolahan Citra *Computer Vision*. *Computer Vision* itu sendiri adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (*Image Processing*) yang memungkinkan komputer dapat melihat seperti manusia. Dengan *vision* tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Pengimplementasian dari *Computer Vision* sebagai berikut *Face Recognition*, *Face Detection*, *Face/Object Tracking*, *Road Tracking*.

OpenCV bersifat gratis baik digunakan untuk penggunaan akademis dan komersial. OpenCV adalah *library Image Processing* berbasis bahasa pemrograman C++ atau Python. OpenCV digunakan oleh banyak *developer*, *scientist*, dan *programmer* untuk mengembangkan sebuah program *pattern recognition*. OpenCV didesain untuk komputasi yang efisiensi, sehingga dapat mempersingkat waktu dalam pembuatan aplikasi yang lumayan rumit (Howse, 2013)

## 2.5 Feature/Keypoint Detector

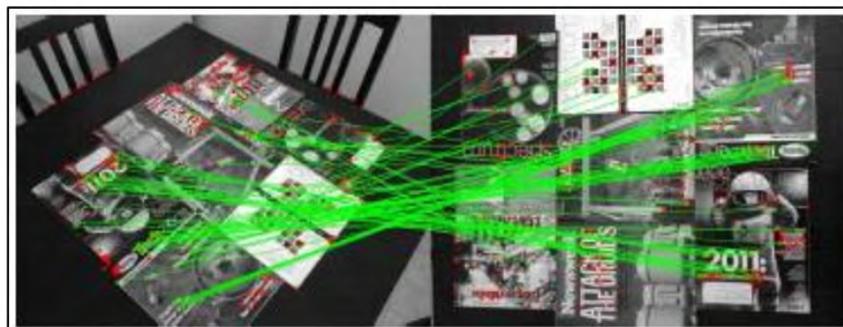
Langkah penting dalam mengenali gambar adalah langkah pencarian feature/keypoint detector dan deskriptornya. Tekstur gambar terdiri dari titik-titik yang memiliki karakteristik unik yang dianggap mampu digunakan sebagai pembeda. Titik unik tersebut disebut keypoint. Setiap keypoint memiliki sifat dan orientasi arah yang disimpan dalam deskriptor dimensi tinggi. Tuytelaar dkk. [4] telah melakukan survey yang lengkap tentang algoritma detektor dan deskriptor. Algoritma yang dikembangkan pada awalnya menggunakan histogram of gradient (HOG) untuk membentuk descriptor. Contoh algoritma yang menggunakan HOG adalah : SIFT, SURF. Algoritma SIFT (scale invariant feature transform) merupakan algoritma pencarian keypoint yang telah lama dikembangkan dan terbukti cukup baik dalam mengenali gambar/citra. Kelemahan utama algoritma ini adalah mahalannya komputasi yang dibutuhkan sehingga tidak cocok untuk diterapkan di perangkat mobile yang sumber dayanya terbatas.[1]

Algoritma dengan HOG membutuhkan konsumsi memori yang cukup besar dan waktu proses yang lama, maka dikembangkan algoritma baru yang berdasarkan pada prinsip binary descriptor. Contoh algoritma dengan binary descriptor adalah : BRIEF, ORB, BRISK, FREAK.

## 2.6 ORB (*Oriented FAST and Rotated BRIEF*)

ORB (*Oriented FAST and Rotated BRIEF*) merupakan deskriptor *binary* yang sangat cepat berbasis BRIEF. ORB memiliki sifat rotasi *invariant* dan tahan terhadap *noise*. Performa ORB dalam uji cobanya menghasilkan efisiensi 2 kali lipat dibandingkan SIFT diberbagai situasi. (Rublee, 2011). Algoritma ORB merupakan kombinasi dari dua teknik yang populer yaitu, FAST (*Features for Accelerated Segment Test*) yang diusulkan oleh Edward Rostenet untuk deteksi key-point dan BRIEF (*Binary Robust Independent Elementary Features*) yang diusulkan oleh Michael Colander untuk deskriptor keypoint. Kedua teknik yakni FAST dan BRIEF sangat atraktif karena memberikan performa yang bagus, kemudian dengan biaya yang lebih murah. Hasil pencocokan menggunakan

ORB dapat dilihat pada Gambar 2.5. Pada gambar tersebut menunjukkan performa ORB dalam mencocokkan citra dengan rotasi yang berbeda.



**Gambar 2.5** Contoh Hasil Pencocokan Menggunakan ORB

(Sumber: Rublee 2011, *ORB: an efficient alternative to SIFT or SURF*)

*Keypoint* FAST merupakan metode yang dipilih untuk menemukan *keypoints* dengan sistem secara *real-time* yang berfungsi untuk melakukan pencocokan (*match*) fitur visual, contohnya seperti *parallel tracking* dan *mapping*. Efisiensi dalam menemukan sudut *keypoints* yang layak dengan beberapa penambahan seperti skema piramida untuk menghitung skala (Rublee, 2011).

Banyak detektor *keypoints* yang menggunakan operator orientasi seperti SIFT dan SURF, tetapi FAST tidak. Deskripsi orientasi dari *keypoints* memiliki banyak cara, ada yang menggabungkan histogram dari komputasi gradient seperti SIFT dan pendekatan berdasarkan *block pattern* pada SURF (Rublee, 2011).

Deskriptor BRIEF merupakan deskriptor fitur yang menggunakan *binary test* yang sederhana antara dua piksel dalam *patch* citra yang sudah dihaluskan (*Smoothed*). Performanya hampir mirip dengan SIFT diberbagai hal termasuk ketahanan untuk pencahayaan, blur, dan *distortion* perspektif. Bagaimanapun deksriptor ini sangat sensitive pada rotasi bidang datar.

Fitur FAST sangat luas digunakan karena sifat komputasinya. Bagaimanapun fitur FAST tidak mempunyai komponen orientasi. oFast (oriented FAST) merupakan perhitungan yang lebih efisien terhadap orientasi yang ditambahkan ke FAST.

FAST mengambil satu parameter yakni intensitas *threshold* antara piksel tengah dengan lingkaran cincin (*ring*) disekitar pusat. FAST tidak menghasilkan

ukuran sudut tapi memiliki respon besar terhadap tepi. Harris *corner* digunakan untuk mengukur *keypoints* yang digunakan. Untuk setiap target nomor N untuk *keypoints* itu diatur dengan *threshold* yang cukup rendah untuk mendapatkan lebih dari N *keypoints*, maka sesuai dengan ukuran Harris dan mengambil N poin teratas. FAST tidak menghasilkan fitur multi-skala, maka dari itu digunakan skala pyramid terhadap citra dan menghasilkan fitur FAST (berdasarkan Harris *filter*) disetiap tingkat piramida. Orientasi terhadap *key-point* menggunakan teknik yakni yang dikenal dengan *intensity centroid*.

Pendekatan berdasarkan orientasi *intensity centroid* menggunakan ukuran sederhana namun efektif dari sudut orientasi dari intensitas centroid. Intensitas centroid mengasumsikan bahwa intensitas sudut yang terkoreksi dari pusat, dan vektor ini dapat digunakan untuk menghubungkan orientasi. Rosin mendefinisikan momen *patch* sebagai berikut:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (2.1)$$

Dimana  $m$  dinotasikan sebagai moment *patch*.  $p$  dan  $q$  adalah urutan momen yang merupakan analog dari momen mekanik,  $x$  dan  $y$  adalah titik koordinat pixel dari image, dan  $I$  dalam persamaan ini adalah nilai inersia rotasi atau massa sudut. Moment tersebut selanjutnya digunakan untuk menemukan *centroid* dengan rumus berikut:

$$C = \left( \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (2.2)$$

Dimana centroid tersebut akan ditemukan dari matriks pixel pada gambar yang didefinisikan sebagai  $m_{00}$ ,  $m_{01}$ , dan sebagainya. Kita bisa membangun sebuah vektor dari pusat sudut,  $\theta$ , ke *centroid*,  $C$ . Secara sederhana orientasi dari patch menjadi sebagai berikut:

$$\theta = \text{atan2}(m_{01}, m_{10}) \quad (2.3)$$

ORB menggunakan deskriptor BRIEF yang berfungsi untuk melakukan deskripsi terhadap bit string dari berbagai *patch* citra yang dibangun dari setiap set

*intensity* tes binari dengan mempertimbangkan citra patch yang diperhalus (*smoothed*),  $p$ . Binari tes  $\tau$  dapat dirumuskan sebagai berikut (Rubble, 2011) :

$$\tau(p; x, y) := f(x) = \begin{cases} 1, & p(x) < p(y) \\ 0, & p(x) \geq p(y) \end{cases} \quad (2.4)$$

Dimana  $x$  dan  $y$  adalah letak titik piksel pada gambar,  $p(x)$  adalah intensitas dari  $p$  pada saat berada di titik  $x$ , dan  $p(y)$  adalah intensitas dari  $p$  pada saat berada di titik  $y$ . Fitur  $f$  digambarkan sebagai vektor dari binari tes  $n$  sebagai berikut :

$$f_n(p) := \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; x, y) \quad (2.5)$$

Dimana:

$f$ : vector dari binary tes

$n$  : binari test

$p$  : patch

$x$  dan  $y$  : titik pixel

## 2.7 PCA (*Principal Component Analysis*)

Pengurangan dimensi (*dimension reduction*) yang terdapat pada sekelompok data digunakan untuk mengurangi beberapa variable yang tidak berelasi antarsatu variabel dengan variabel lainnya dalam satu kelompok klaster. Tujuan dari pengurangan dimensi adalah untuk mendapatkan variabel – variabel yang optimal yang dapat membentuk klaster yang diinginkan. Salah satu metode pengurangan dimensi adalah *Principal Component Analysis*.

*Principal Component Analysis* menurut Lindsay (2002) adalah sebuah metode untuk mengidentifikasi pola – pola yang terdapat dalam sebuah data dan menyatakannya dalam sebuah cara untuk menentukan kemiripan dan perbedaan yang dimiliki oleh data tersebut. Salah satu keunggulan yang dapat ditemukan dalam

*Principal Component Analysis* adalah dengan melakukan metode ini dapat mengurangi jumlah dimensi yang terdapat dalam satu pola tanpa mengurangi informasi yang terdapat dalam data tersebut. Oleh karena itu, *Principal Component Analysis* sangat diperlukan untuk membantu reduksi terhadap pola – pola yang ada dalam suatu klaster.

*Principal Component Analysis* sangat cocok untuk digunakan terhadap *high dimensional dataset*. PCA lebih dekat dikenal dengan analisa faktor (factor analysis). Beberapa langkah yang digunakan untuk menggunakan metode *Principal Component Analysis* adalah :

- a) Persiapkan data yang akan dianalisa dengan menggunakan *Principal Component Analysis*.
- b) Hitung *mean* untuk kelompok data tersebut.
- c) Melakukan perhitungan untuk matrik kovarian, dimana varian dihitung sebagai :

$$S^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{(n-1)} \quad (2.6)$$

Dimana :

$S^2$  : kovarian data

$n$  : jumlah data

$X_i$  : data ke-i

$\bar{X}$  : rata – rata semua data

Dan kovarian dihitung sebagai :

$$Cov(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)} \quad (2.7)$$

Dimana :

$X_i$  : koordinat X data ke-i

$\bar{X}$  : rata – rata X semua data

$Y_i$  : koordinat Y data ke-i

$\bar{Y}$  : rata – rata Y semua data

Kovarian matrik dihitung sebagai:

Jika kita memiliki data dengan dimensi  $(x, y, z)$ , kita dapat melakukan kalkulasi  $Cov(X,Y)$  ,  $Cov(X,Z)$  ,  $Cov(Y,Z)$  . Dengan kata lain, apabila kita memiliki data dengan dimensi  $n$ , maka banyaknya kovarian matrik yang kita hitung yaitu

$$\frac{n!}{(n-2)! \times 2}$$

- d) Melakukan perhitungan eigenvectors dan eigenvalues dari matrik kovarian.
- e) Mendapatkan dataset yang baru berdasarkan variabel – variabel yang terkait dalam satu faktor.

## 2.8 K-Nearest Neighbor (KNN)

Algoritma K-Nearest Neighbor (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap obyek berdasarkan data pembelajaran yang jaraknya paling dekat dengan obyek tersebut. Prinsip kerja dari K-Nearest Neighbor (KNN) adalah mencari jarak terdekat antara data yang akan dievaluasi dengan K tetangga (neighbor) terdekatnya dalam data pelatihan.

Pada fase pembelajaran, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi dari data pembelajaran. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk data test (yang klasifikasinya tidak diketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor data pembelajaran dihitung, dan sejumlah k buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut.

Nilai k yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan cross-validation. Kasus khusus di mana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat

(dengan kata lain,  $k = 1$ ) disebut algoritma k-nearest neighbor. Ketepatan algoritma k-NN ini sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur agar performa klasifikasi menjadi lebih baik.

Sesuai dengan prinsip kerja K-Nearest Neighbor yaitu mencari jarak terdekat antara data yang akan dievaluasi dengan k tetangga(neighbor) terdekatnya dalam data pelatihan. Persamaan dibawah ini menunjukkan rumus perhitungan untuk mencari jarak terdekat dengan  $d$  adalah jarak dan  $p$  adalah dimensi data (Agusta, 2007):

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2} \quad (2.8)$$

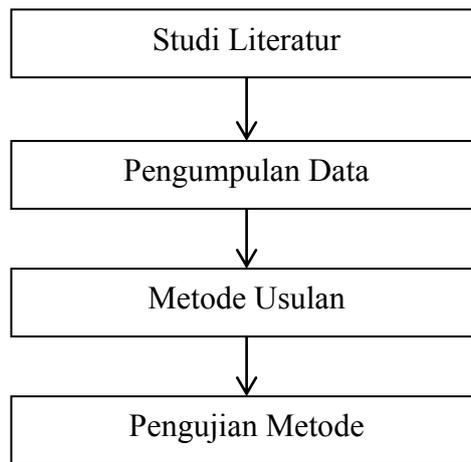
Dengan keterangan :

<b>x</b> :sampel data	<b>p</b> :dimensi data
<b>d</b> :jarak	<b>i</b> :variable data
<b>y</b> :data uji	

## **BAB III**

### **METODE PENELITIAN**

Tahapan-tahapan yang dilalui pada penelitian ini meliputi (1) Studi Literatur, (2) Pengumpulan Data, (3) Metode Usulan, (4) Pengujian Metode, Gambar 3.1 menunjukkan tahapan-tahapan tersebut secara berurut.



**Gambar 3.1** Tahapan Penelitian

#### **3.1 Studi Literatur**

Studi literatur terhadap penelitian ini adalah dengan memahami berbagai ragam perbedaan dari bentuk jenis ikan, baik berupa warna, bentuk, maupun polanya. Setiap jenis ikan mempunyai beragam perbedaan terutama bentuk badan, kepala, bentuk sirip, dan lain-lain sesuai habitatnya. Kemudian mempelajari berbagai metode yang berhubungan dengan klasifikasi objek.

#### **3.2 Pengumpulan Data**

Dalam penelitian ini, dataset yang digunakan yaitu dataset dari Robotics@QUT Channel. Data berupa file citra dengan ekstensi JPG dengan ruang warna RGB. Dataset ikan ini terdiri dari 165 gambar yang dikumpulkan dari 15 jenis ikan. Tahapan pada penelitian ini tidak memerlukan tahapan *pre-processing* untuk mensegmentasi citra, karena data yang digunakan pada penelitian ini juga akan diuji dengan background yang

berbeda untuk mengenali fitur yang dihasilkan. Contoh sample data yang akan digunakan pada penelitian ini dapat dilihat pada gambar 3.2.



**Gambar 3.2** *Sample* Data Pengujian

### 3.3 Metode Usulan

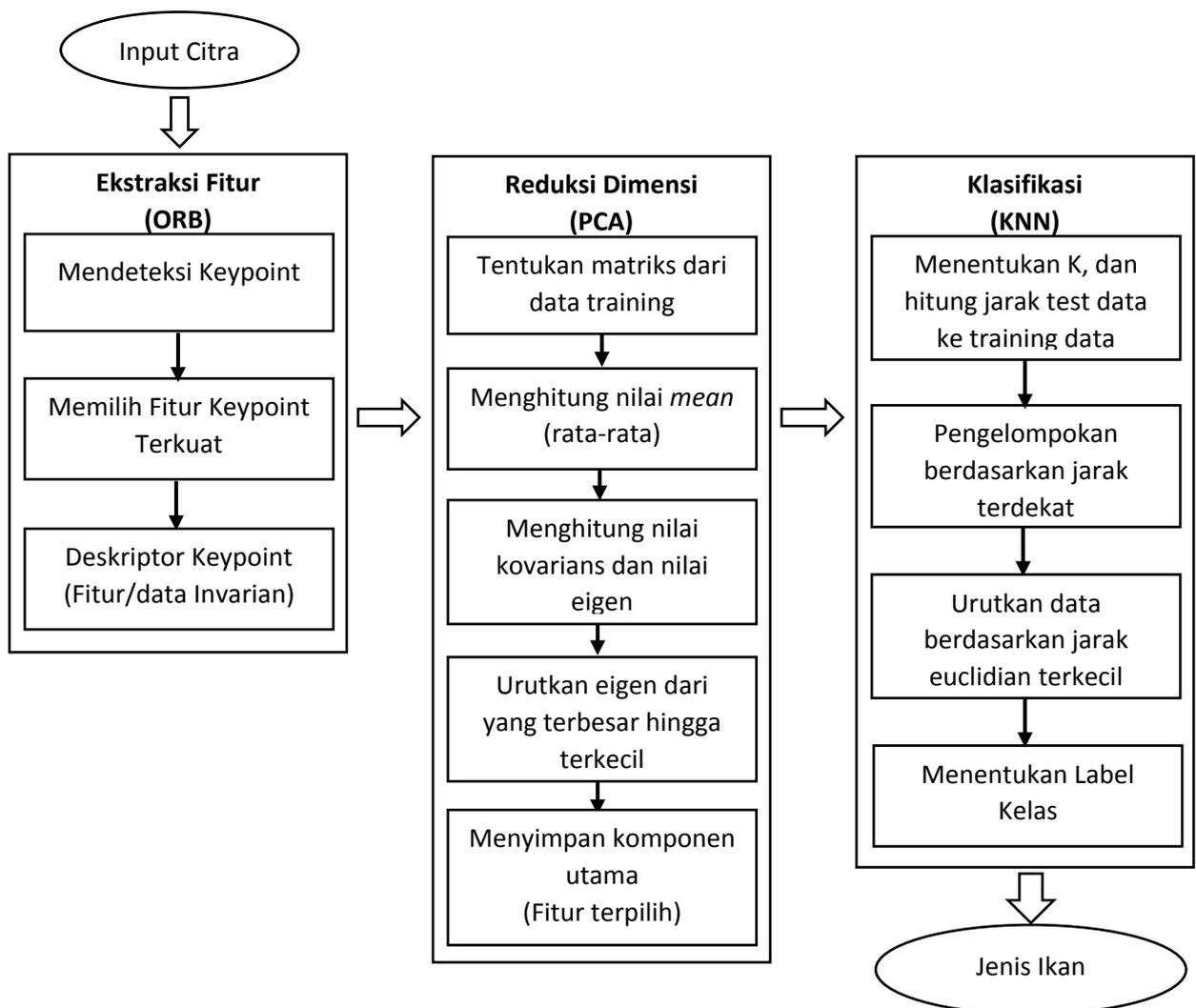
Bagian ini menyajikan tentang berbagai langkah yang terkait dengan metode penelitian kami. Tahapan dalam penelitian ini bisa dilihat pada Gambar 3.3



**Gambar 3.3** Metode yang diusulkan

Tahap awal dari metode ini adalah input gambar ikan dari dataset yang disediakan. Data ikan diproses dengan invariant, artinya data ikan yang akan dikenali tidak bergantung pada ukuran maupun warna pada image gambar. Citra input yang diproses sudah memiliki hasil segmentasi citra, sehingga dalam metode ini tidak memerlukan tahap *pre-processing* untuk memisahkan objek dengan background. Citra dari input akan melalui proses ekstraksi fitur menggunakan metode ORB, metode ini akan menghasilkan suatu citra dengan keypoint (fitur lokal) yang invariant terhadap rotasi. Fitur hasil dari ekstraksi tersebut kemudian disertai dengan pengurangan dimensi dengan menggunakan metodologi PCA. Metode PCA ini akan menghasilkan keypoint

atau fitur setelah dilakukan pengurangan dimensi dengan tetap mempertahankan informasi penting yang ada. Kemudian, deskriptor dari kedua gambar masukan dan gambar pada database akan dicocokkan dengan metode KNN (*K-Nearest Neighbor*) untuk mengklasifikasi jenis ikan dari gambar masukan. Metode yang diusulkan akan dapat membedakan jenis ikan dari bentuk tubuh ikan yang bervariasi secara invarian. Perbedaan tubuh ikan tersebut dipengaruhi oleh bentuk badan, bentuk sirip, dan ekor ikan. Pada Gambar 3.4 merupakan gambaran detail proses dari metode yang diusulkan.



**Gambar 3.4** Gambaran detail proses metode yang diusulkan

### 3.3.1 *ORB (Oriented FAST Rotated BRIEF) Detector*

ORB merupakan kombinasi dari dua teknik, yaitu FAST dan BRIEF. Keduanya dipilih karena kinerjanya dan biaya rendah, dan karakteristik tertentu yang diinginkan seperti invarian mereka terhadap iluminasi, kabur, affine dan sebagainya. ORB mengatasi beberapa kekurangan penting dari mekanisme yang ada sebelumnya seperti kurangnya komponen orientasi FAST dan kurangnya rotasi invarian pada BRIEF. Ini beroperasi dengan menambahkan komponen orientasi yang akurat ke FAST dengan memanfaatkan mekanisme Intensity Centroid Cloud dan membuat rotasi BRIEF yang invarian dengan membangun varian yang disebut steered BRIEF dan kemudian mengembangkannya menjadi r-BRIEF offstring, yang merupakan rotasi yang invarian.

Orientasi Key-Point FAST yang dimodifikasi disebut oFAST dan diimplementasikan dengan mendeteksi titik FAST pada gambar yang diberikan dan karena parameter yang dipertimbangkan FAST adalah intensitas threshold antara piksel tengah dan piksel pada lingkaran pusat (FAST-9 dengan radius melingkar 9 diikutsertakan di sini karena hal ini terbukti menguntungkan karena peningkatan kinerja yang ditawarkannya). FAST tidak memiliki kemampuan untuk memberikan ukuran sudut pandang namun cukup responsif sepanjang tepinya dan karenanya ukuran Harris Corner dapat digunakan untuk mendapatkan key-point FAST secara tepat. Untuk mencapai hal ini, mari kita anggap kita memiliki target titik kunci  $N$ , maka kita perlu menetapkan Threshold dengan cukup rendah sedemikian rupa sehingga kita mendapatkan lebih dari  $N$  poin kunci, yang kemudian dapat dipesan sesuai dengan Ukuran Hessian dan akhirnya kita bisa memilih titik  $N$  teratas. Lebih jauh lagi, karena FAST tidak menghasilkan jenis yang multi skala, piramida skala gambar digunakan untuk menghasilkan fitur FAST yang cukup disaring menggunakan ukuran Harris di setiap tingkat di piramida.

Komponen Orientasi ditambahkan pada FAST dengan menggunakan pendekatan Intensity Centroid (IC) yang menggunakan ukuran kuat pada orientasi sudut. Sebuah patch yang digunakan untuk menemukan centroid telah direpresentasikan pada persamaan 2.1. Kemudian dengan mempertimbangkan momen pada persamaan 2.1, sentroid akan diperoleh melalui persamaan 2.2. Sebuah vektor dibangun dari pusat ke sentroid OC dan kemudian disesuaikan dengan orientasi patch melalui persamaan 2.3.

Mengingat pentingnya parameter iluminasi dari sudut tidak diperhitungkan karena ukuran sudut tetap sama terlepas dari jenis sudutnya. Invarian rotasi dapat ditingkatkan dengan memastikan bahwa momen dihitung terhadap  $x$  dan  $y$  yang berada di dalam daerah lingkaran radius  $r$ . Pilihan optimal untuk ukuran patch adalah  $r$ , dengan cara yang memastikan bahwa jarak dari  $x, y$  adalah dari  $[-r, r]$ . Dengan ukuran Hessian, karena nilai  $C$  mendekati nol, maka cenderung menjadi tidak stabil namun ini tidak terjadi dengan sudut FAST, yang menguntungkan bagi efisiensi sistem.

Kedua, ORB melibatkan penambahan komponen sadar rotasi yang disebut r-BRIEF yang merupakan versi evolusi dari *steered BRIEF Descriptor* yang dikemudian ditambah dengan langkah pembelajaran terkait juga diuraikan dalam [1] untuk menemukan fitur biner yang kurang berkorelasi.

Untuk memastikan rotasi yang efisien dari operator BRIEF, deskripsi bit string dari patch citra dibangun dari satu set tes intensitas biner. Untuk mengilustrasikan operasi dari BRIEF konvensional yang lebih baik, sebelum menambahkan komponen orientasi oleh ORB, mari kita anggap ada yang merapikan patch gambar  $p$ . Kemudian untuk tes binernya direpresentasikan sebagai berikut:

$$\tau(p; x, y) := \begin{cases} 1 & : p(x) < p(y) \\ 0 & : p(x) \geq p(y) \end{cases} \quad (3.1)$$

dimana  $p(x)$  menunjukkan intensitas patch menutupi titik yang diberikan  $x$ , dan  $p(y)$  menunjukkan intensitas patch menutupi titik yang diberikan  $y$ .

Selanjutnya, fitur yang merupakan fungsi Dari patch yang dianggap kemudian dapat direpresentasikan sebagai vektor Dari uji biner  $n$  sebagai berikut:

$$f_n(p) := \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; x_i, y_i) \quad (3.2)$$

Dalam pertimbangan kami, kami menggunakan distribusi Gaussian di sekitar pusat patch beserta pilihannya. Alasan kenaikan kinerja tambahan yang disodorkan dengan metode yang kami ajukan. Karena, salah satu kontribusi penting ORB adalah inversi rotasi pada BRIEF, karena BRIEF mengalami penurunan tajam dalam adanya rotasi yang melebihi beberapa derajat. Langkah pertama dalam pendekatan mereka adalah

mengarahkan BRIEF sesuai dengan orientasi poin-poin utama (langkah ini dijuluki *steered-BRIEF*).

*Steered BRIEF* dilakukan sebagai berikut:

Kita tentukan untuk rangkaian fitur tertentu dari tes biner  $n$  di lokasi tertentu  $(x_i, y_i)$ , matriks  $2 \times n$  dapat digambarkan sebagai berikut:

$$S = \begin{pmatrix} x_1, \dots, x_n \\ y_1, \dots, y_n \end{pmatrix} \quad (3.3)$$

Dimana,  $S$  merupakan nilai *Steered*, dan  $(x, y)$  adalah lokasi koordinat.

Selanjutnya, dengan memanfaatkan  $\theta$  (orientasi patch) dan  $R_\theta$  (matriks rotasi yang sesuai), versi *steered*  $S_\theta$  dari  $S$  dapat diperoleh sebagai berikut:

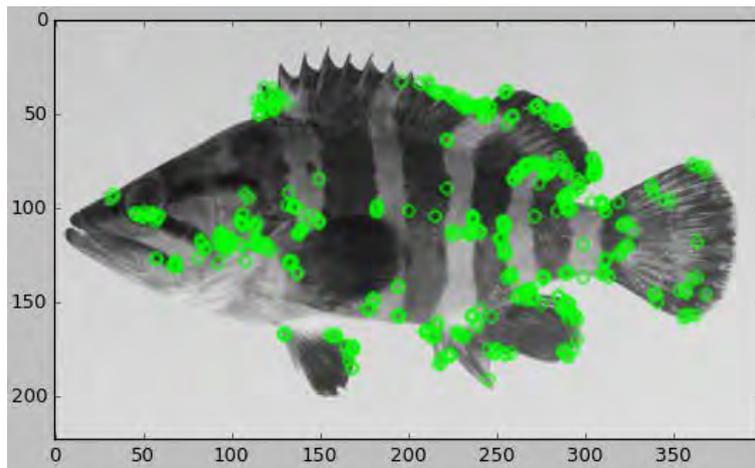
$$S_\theta = R_\theta S \quad (3.4)$$

Sehingga, operator *steered BRIEF* dapat ditulis sebagai:

$$g_n(p, \theta) := f_n(P) | (x_i, y_i) \in S_\theta \quad (3.5)$$

Kami mendiskretkan sudut ke tahapan dari  $\frac{2\pi}{30}$  (12 derajat) dan membangun tabel pencarian dari pola *pre-computed BRIEF*. Selama orientasi key-point  $\theta$  konsisten di seluruh tampilan, rangkaian poin  $S_\theta$  yang benar akan digunakan untuk menghitung deskriptornya.

Selanjutnya, metode ini akan memperoleh deskriptor keypoint yang invariant terhadap perubahan intensitas cahaya atau perubahan sudut pandang tiga dimensi. Sehingga output yang dihasilkan yaitu berupa citra ikan yang telah terdeteksi dan keypoint/fitur pada ikan tersebut yang akan dicocokkan dengan ikan yang paling mirip dengan jenis ikan yang ada pada dataset. Contoh hasil output dari metode ORB dapat dilihat pada Gambar 3.5. Gambar tersebut menunjukkan citra ikan yang telah terdeteksi beserta deskriptor keypoint yang mencirikan dari jenis ikan tersebut.



**Gambar 3.5** Hasil deteksi keypoint menggunakan ORB

Hasil output dari metode ORB tidak hanya menghasilkan keypoint saja. Fitur yang dihasilkan melalui ORB ini sebanyak 500 fitur dengan nilai ukuran dimensi sebesar 32. Fitur tersebut telah di deskripsikan, sehingga invarian terhadap rotasi pada citra.

### 3.3.2 PCA (*Principal Component Analysis*)

Analisis komponen utama adalah prosedur statistik yang diterapkan untuk mengurangi dimensi data. PCA beroperasi dengan mengubah satu set data masukan dari dimensi yang lebih tinggi ke data dimensi yang lebih rendah. Di sini, masukan ke PCA adalah deskriptor ORB dari keduanya, *data testing* dan *data training*. Vektor eigen Dari matriks korelasi ditemukan dan hanya nilai Eigen yang secara signifikan lebih tinggi yang dipertimbangkan. Ini digambarkan sebagai transformasi dari himpunan vektor masukan  $N$  yang diberikan dengan panjang yang sama  $K$  ( $=32$  untuk deskriptor ORB) pada vektor  $N$ -dimensi  $X = [x_1, x_2, \dots, x_N]^T$  ke dalam vektor  $y$  yang diberikan oleh

$$y = A(x - m_x) \tag{3.6}$$

Dimana,  $A$  merupakan vector Eigen, setiap baris  $x$  terdiri dari nilai  $K$  yang termasuk dalam satu input. Vektor  $m_x$  pada persamaan (3.6) adalah vektor *mean*, dan secara matematis diberikan sebagai

$$m_x = E\{x\} = \frac{1}{K} \sum_{k=1}^K x_k \quad (3.7)$$

Matriks A di Persamaan (3.6) ditentukan dari matriks kovariansi  $C_x$ . Baris dalam matriks A terdiri dari vektor eigen dari  $C_x$  yang dipesan sesuai dengan nilai Eigen yang sesuai dalam urutan menurun. Matriks  $C_x$  diperoleh dengan relasi yang diberikan oleh

$$C_x = E\{(x - m_x)(x - m_x)^T\} = \frac{1}{K} \sum_{k=1}^K x_k x_k^T - m_x m_x^T \quad (3.8)$$

Dimana  $C_x$  merupakan Nilai Matriks Kovarians, dan  $E$  adalah nilai Eigen. Ukuran matriks kovariansi adalah  $N \times N$  karena  $x$  adalah  $N$ -dimensi. Kovariansi antara vektor input  $x_i, x_j$  diberikan oleh

$$C_x(i, j) = E\{(x_i - m_i)(x_j - m_j)\} \quad (3.9)$$

Sebagai tambahan, agar PCA dapat bekerja dengan tepat, harus menggunakan data standar sehingga rata-rata adalah nol dan estimasi varians yang tidak bias adalah satu kesatuan. PCA mengasumsikan bahwa data mengikuti distribusi *Gaussian*. Meskipun PCA tampaknya efektif dalam penerapannya, hal itu dibatasi dengan dimensi. Jumlah komponen yang dapat dipertahankan dapat ditentukan oleh relasi

$$\frac{\sum_{i=1}^N V_i}{\sum_{k=1}^K V_k} = \alpha \quad \text{where } 0 < \alpha \leq 1 \quad (3.10)$$

Dalam persamaan di atas,  $V$  mewakili nilai Eigen,  $K$  mewakili jumlah komponen utama yang dipertahankan, dan  $N$  adalah total nilai Eigen. Komponen yang dipertahankan melalui metode PCA ini berupa fitur penting dari citra ikan yang telah terdeteksi melalui ORB. Ukuran dimensi dari hasil ekstraksi fitur menggunakan ORB akan dikurangi. Pengurangan dimensi ini ditujukan untuk menyisihkan beberapa dimensi fitur yang tidak penting, agar tidak membebani proses klasifikasi menggunakan metode K-Nearest Neighbor.

### 3.3.3 Klasifikasi Menggunakan *K-Nearest Neighbor* (KNN)

Dalam Tahap ini adalah proses pengenalan menggunakan algoritma *K-Nearest Neighbor* (KNN). Algoritma ini dioptimalkan untuk pencarian tetangga terdekat yang cepat untuk kumpulan data besar. Algoritma ini sangat rentan terhadap dimensionalitas yang tinggi. Namun permasalahan telah diatasi dengan menurunkan dimensionalitas menggunakan PCA. Dengan demikian, hal pertama yang akan dilakukan adalah memasukkan input berupa hasil rekayasa fitur yang dihasilkan oleh PCA. Proses pencarian ketetanggan terdekat dalam tahapan ini yaitu menggunakan persamaan yang telah dijelaskan pada persamaan 2.8.

Hasil dari tahapan ini yaitu berupa rank ketetanggan terdekat dengan nilai tertinggi hingga terendah. Label kelas yang paling tepat akan ditentukan berdasarkan jumlah kemunculan label kelas dan yang tertinggi pada urutan rank. Hasil akhir dari metode ini berupa citra ikan dengan label kelas berupa nama jenis ikan yang telah terdeteksi.. Hasil akhir dari metode ini dapat dilihat pada gambar 3.6.



**Gambar 3.6** Hasil klasifikasi ikan menggunakan KNN

Seperti yang ditunjukkan pada Gambar 3.6, output yang dihasilkan yaitu berupa gambar input yang ditambahkan nama kelas yang tepat pada citra. Label kelas ditentukan berdasarkan jenis ikan yang ditampilkan terbanyak atau tertinggi dari rank ketetanggan terdekat.

### **3.4 Analisis Hasil**

Proses pengujian metode yang diusulkan akan dilakukan pada perangkat dengan prosesor Intel® Core™ i5 CPU M450 @ 2.40 GHz, memori 4 GB, dan sistem operasi

yang digunakan adalah Windows 10 Home 64-bit. Metode yang diusulkan akan dapat mengklasifikasi jenis ikan secara invariant, tidak bergantung pada ukuran, atau rotasi

Untuk menguji tingkat akurasi dari hasil klasifikasi dari metode yang kami usulkan. Hasil dari penelitian ini juga akan dilakukan perbandingan dengan metode algoritma ORB-KNN, BRISK-PCA-KNN, dan BRISK-KNN. Dari hasil perbandingan tersebut, maka akan dilakukan analisa untuk menemukan metode yang terbaik dalam klasifikasi ikan.

## BAB 4

### UJI COBA DAN ANALISA HASIL

Berdasarkan skenario dan langkah-langkah yang telah dijelaskan pada Bab 3, maka pada Bab ini akan dibahas hasil uji coba dan analisa hasil.

#### 4.1 Lingkungan Uji Coba

Sebagai uji coba pada penelitian ini, data sampel diujikan dengan menggunakan komputer (laptop) dengan dukungan processor Intel® Core™ i5 CPU M450 @ 2.40 GHz, kapasitas memory 4GB. Perangkat lunak pendukung adalah sistem operasi windows 10 pro, bahasa pemrograman OpenCV versi 3.3, C++, serta visual studio 2015.

#### 4.2 Data Uji Coba

Uji coba algoritma klasifikasi pada penelitian ini menggunakan 165 data citra ikan dari Robotics@QUT. Dataset tersebut akan dibagi menjadi 40 data *testing* dan 125 data *training* dengan 15 jenis ikan yang berbeda. Ukuran piksel pada setiap ikan bervariasi, sehingga data input tersebut akan diproses secara invarian. Data citra diambil dengan latar belakang yang berbeda. Citra data *testing* dalam penelitian ini dapat dilihat pada gambar 4.1.



**Gambar 4.1** Data *testing*



### 4.3.1 Pengujian Ekstraksi Fitur

Pengujian ekstraksi fitur pada penelitian ini dilakukan untuk menghasilkan *interest point* yang merupakan fitur penting pada data *testing* untuk kemudian diproses dengan cara dibandingkan dengan fitur pada semua data *training*. Proses dari pengujian ini menggunakan metode ORB. Pada tahap pengujian ini, citra input adalah berupa citra ikan dari data *testing* dan data *training*. Gambar citra input dapat dilihat pada gambar 4.3



(a) Citra testing



(b) Citra training 1



(c) Citra training 2

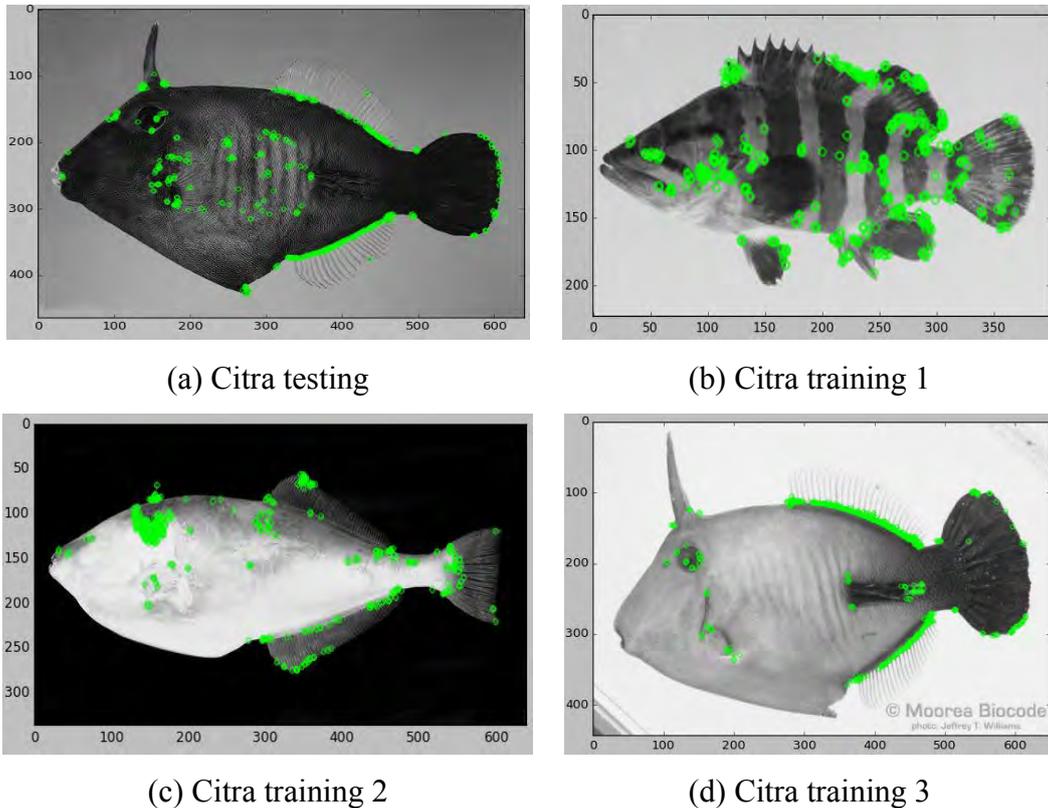


(d) Citra training 3

**Gambar 4.3** Citra input

Gambar input yang dimasukkan yaitu berupa 1 citra *testing*, dan 3 citra *training* dengan salah satu jenis ikan yang sama. Hasil dari pengujian ekstraksi fitur ini yaitu berupa keypoint penting yang telah terdeteksi dan berupa 500 fitur yang telah dideskripsikan di setiap gambar input. Keypoint akan terdeteksi dari beberapa bagian ikan yang menjadi suatu ciri khas dari masing-masing citra tersebut. Setelah keypoint terdeteksi, maka langkah selanjutnya dari ORB adalah mendeskripsikan berbagai ciri tersebut menjadi fitur yang tahan terhadap perubahan rotasi. Fitur yang telah

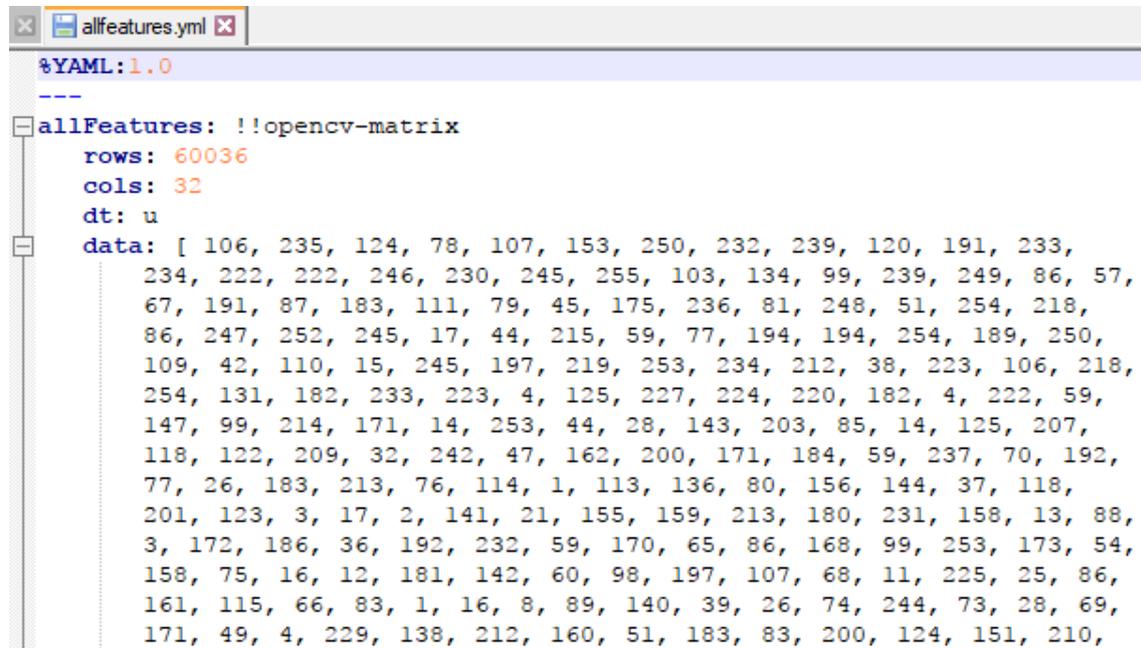
dideskripsikan akan disimpan berupa data matriks. Data matriks tersebut kemudian akan diproses sebagai input untuk proses reduksi dimensi atau langsung proses klasifikasi. Keypoint penting yang telah terdeteksi menggunakan metode ORB dapat dilihat pada gambar 4.4.



**Gambar 4.4** Keypoint yang terdeteksi menggunakan ORB

Seperti yang ditunjukkan pada gambar 4.4, titik yang berwarna hijau merupakan kumpulan keypoint penting yang telah terdeteksi dari masing-masing ikan. Gambar (b) dan (c) merupakan jenis ikan yang berbeda dengan jenis ikan pada citra testing (a), sedangkan Gambar (d) merupakan jenis ikan yang sama seperti gambar (a), sehingga memiliki susunan keypoint yang mirip dan berdekatan. Setelah *keypoint* pada setiap citra terdeteksi, maka semua fitur yang didapatkan tersebut akan dideskripsikan dan disimpan dalam bentuk data matriks untuk kemudian diproses melalui PCA. Deskripsi point dari data training disimpan untuk proses lanjutan PCA. Data matriks yang dihasilkan oleh data training tersebut dapat dilihat pada gambar 4.5. Data-data yang ada

pada fitur tersebut merupakan nilai gradien pada histogram tingkat keabuan mulai dari 0-255 di sekitar *interest point* pada citra yang disebut *keypoints*.



```
%YAML:1.0
---
allFeatures: !!opencv-matrix
  rows: 60036
  cols: 32
  dt: u
  data: [ 106, 235, 124, 78, 107, 153, 250, 232, 239, 120, 191, 233,
    234, 222, 222, 246, 230, 245, 255, 103, 134, 99, 239, 249, 86, 57,
    67, 191, 87, 183, 111, 79, 45, 175, 236, 81, 248, 51, 254, 218,
    86, 247, 252, 245, 17, 44, 215, 59, 77, 194, 194, 254, 189, 250,
    109, 42, 110, 15, 245, 197, 219, 253, 234, 212, 38, 223, 106, 218,
    254, 131, 182, 233, 223, 4, 125, 227, 224, 220, 182, 4, 222, 59,
    147, 99, 214, 171, 14, 253, 44, 28, 143, 203, 85, 14, 125, 207,
    118, 122, 209, 32, 242, 47, 162, 200, 171, 184, 59, 237, 70, 192,
    77, 26, 183, 213, 76, 114, 1, 113, 136, 80, 156, 144, 37, 118,
    201, 123, 3, 17, 2, 141, 21, 155, 159, 213, 180, 231, 158, 13, 88,
    3, 172, 186, 36, 192, 232, 59, 170, 65, 86, 168, 99, 253, 173, 54,
    158, 75, 16, 12, 181, 142, 60, 98, 197, 107, 68, 11, 225, 25, 86,
    161, 115, 66, 83, 1, 16, 8, 89, 140, 39, 26, 74, 244, 73, 28, 69,
    171, 49, 4, 229, 138, 212, 160, 51, 183, 83, 200, 124, 151, 210,
```

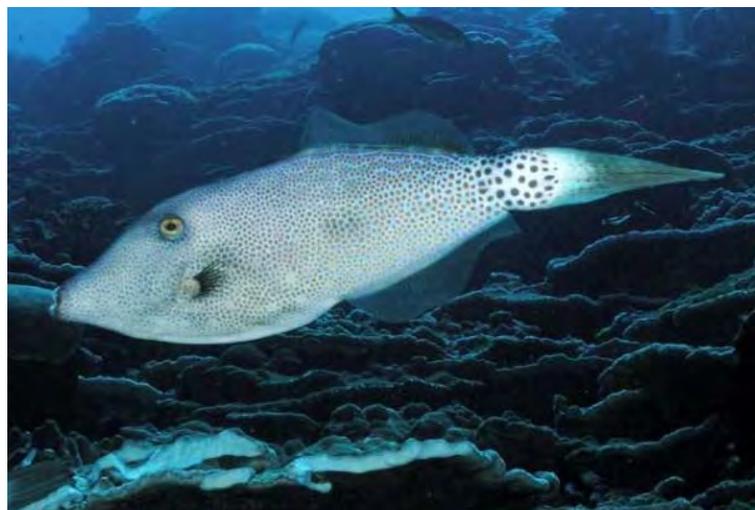
**Gambar 4.5** Matriks hasil deskripsi fitur ORB

Seperti yang ditunjukkan pada gambar 4.5, Matriks yang dihasilkan setelah deskripsi fitur dari 125 data training tersebut adalah sebanyak 60036 baris, dan 32 kolom. Matriks tersebut akan dijadikan input untuk proses reduksi dimensi menggunakan PCA sebagai proses reduksi dimensionalitas untuk mengurangi ukuran dimensi dari fitur dan merekayasa fitur yang dihasilkan. Nilai matriks tersebut akan menghasilkan nilai eigen yang kemudian diproses untuk setiap data *testing*. Rasio PCA yang digunakan dalam pengujian ini adalah 90%. Analisa pengaruh PCA dalam klasifikasi ikan akan dijelaskan setelah semua tahapan pengujian telah dilakukan. Kemudian proses klasifikasi akan dilakukan dengan menghitung semua keypoint atau fitur yang terdeteksi pada semua dataset untuk menentukan label kelas yang paling tepat berdasarkan urutan *rank* nilai ketetanggan terdekat dari kumpulan keypoint atau fitur yang dideteksi.

### 4.3.2 Pengujian Metode Yang Diusulkan

Setelah tahap ekstraksi fitur selesai, tahap selanjutnya adalah mereduksi dimensi dari deskripsi fitur yang dihasilkan menggunakan *library* PCA pada *opencv*. Input dari proses reduksi dimensi ini adalah jumlah baris dan kolom pada matriks dari data fitur. Seperti yang ditunjukkan pada gambar 4.5, jumlah baris dan kolom yang dijadikan input adalah sebanyak 60036 baris, dan 32 kolom. Dan hasil dari reduksi dimensi tersebut yaitu merupakan rekayasa fitur terbaru dengan nilai dimensi yang lebih rendah. Kemudian fitur yang direkayasa tersebut akan diproses menggunakan metode KNN. Dimana dari fitur KNN ini akan dicari nilai ketetanggaan yang paling terdekat dan yang paling banyak kecocokan diantara 500 fitur yang dihasilkan dari data *testing* dan data *training*.

Sebelum melakukan pengujian pada keseluruhan data, maka perlu menentukan nilai ketetanggaan yang tepat. Menentukan nilai K sangat penting dalam proses klasifikasi menggunakan KNN. Untuk membahas dampak dari nilai k yang berbeda untuk hasil klasifikasi, kami secara acak memilih tiga kelompok dengan k yang berbeda,  $k = 5$ ,  $k = 8$  dan  $k = 10$ . Nilai k dari penelitian ini akan berpengaruh terhadap jumlah *rank* ketetanggaan terdekat untuk kemudian memilih label kelas yang tepat dari semua rank tersebut. Sehingga pemilihan k yang tepat sangat berpengaruh untuk pengambilan keputusan label kelas yang akan ditentukan. Penentuan nilai k ini akan diuji coba dengan cara menginputkan satu citra ikan, jenis ikan yang diinputkan dalam penentuan k ini adalah ikan jenis *aluterus scriptus*, seperti yang ditampilkan pada gambar 4.6.



**Gambar 4.6** Citra input data pengujian penentuan nilai k

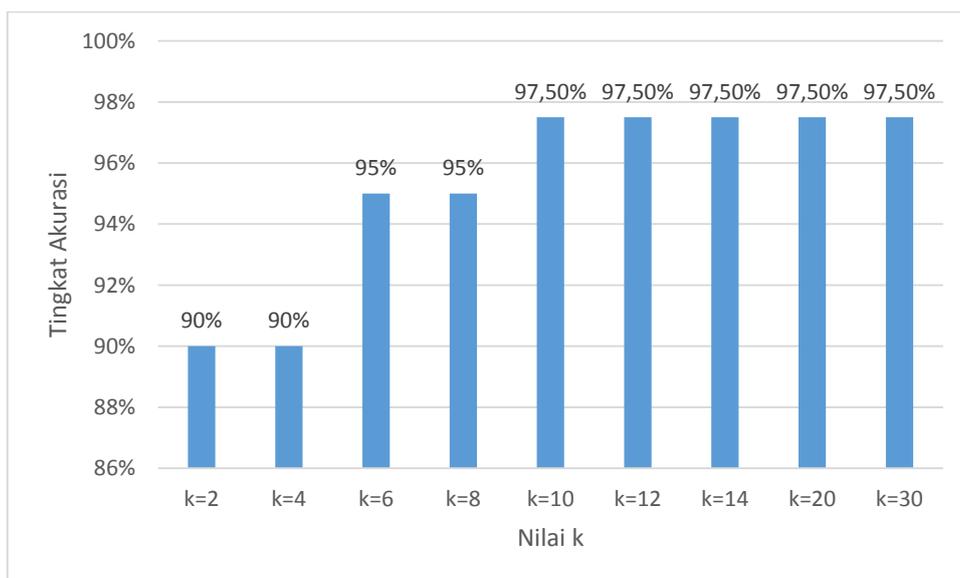
Setelah menginputkan citra ikan seperti gambar 4.6., maka algoritma KNN akan menampilkan urutan rank dengan jumlah yang sesuai dengan nilai k yang ditentukan.

Tabel 4.1 Hasil komparatif penentuan nilai K

	K=5	K=8	K=10
Rank 1	aluterus_scriptus	aluterus_scriptus	aluterus_scriptus
Rank 2	acanthistius_cinctus	acanthistius_cinctus	acanthistius_cinctus
Rank 3	halichoeres_chrysus	halichoeres_chrysus	halichoeres_chrysus
Rank 4	bothus_pantherinus	bothus_pantherinus	bothus_pantherinus
Rank 5	aluterus_monoceros	aluterus_monoceros	aluterus_monoceros
Rank 6	-	halichoeres_chrysus	halichoeres_chrysus
Rank 7	-	aluterus_scriptus	aluterus_scriptus
Rank 8	-	aluterus_monoceros	aluterus_monoceros
Rank 9	-	-	aluterus_scriptus
Rank 10	-	-	aluterus_scriptus

Dapat dilihat dari tabel 4.1, penentuan nilai  $k=5$  hanya menghasilkan 5 rank dengan 5 label kelas jenis ikan yang berbeda-beda, sehingga kita tidak bisa menentukan jenis ikan yang tepat dari 5 rank tersebut, karena tidak menampilkan label kelas yang berulang. Sama halnya dengan nilai  $k=8$ , dari 8 rank tersebut menampilkan ada dua label kelas yang ditampilkan secara berulang, yaitu aluterus\_scriptus dan aluterus\_monoceros. Sehingga, nilai  $k=8$  ini juga masih belum cukup untuk menentukan jenis ikan yang tepat karena ada dua label kelas terbaik yang ditampilkan sebanyak dua kali dalam urutan rank (label kelas berwarna merah dan biru), sehingga jenis ikan yang tepat belum bisa ditentukan. Hasil penentuan nilai k terbaik yaitu  $k=10$ . Pengujian dengan penentuan nilai  $k=10$  ini menghasilkan label kelas yang ditampilkan berulang sebanyak 4 kali dalam 10 rank, yaitu label kelas aluterus\_scriptus. Sehingga,

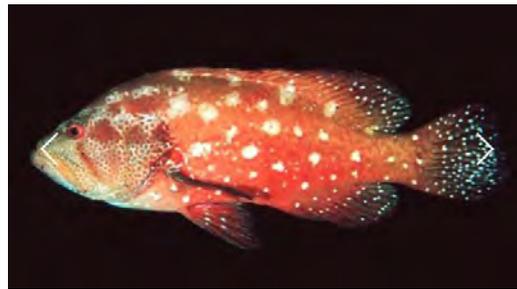
dengan menentukan nilai  $k=10$ , kami bisa mendapatkan label kelas jenis ikan yang lebih akurat. Pengujian nilai  $k$  ini juga dibuktikan melalui hasil eksperimen dengan memasukkan nilai  $k$  secara acak. Nilai  $k$  yang diujicoba adalah 2, 4, 6, 8, 10, 12, 14, 20, 30 dan seterusnya. Penentuan nilai  $k$  yang tepat akan menghasilkan nilai akurasi tertinggi. Hasil dari percobaan ini dapat dilihat pada grafik yang ditampilkan pada gambar 4.7



**Gambar 4.7** Perbandingan tingkat akurasi setiap nilai  $k$

Gambar 4.7 menunjukkan bahwa hasil akurasi tertinggi dicapai dengan nilai  $k \geq 10$ , nilai  $k$  yang digunakan adalah nilai  $k$  minimal dari akurasi tertinggi. Sehingga nilai  $k$  paling tepat adalah 10.

*Rank* citra yang ditampilkan pada setiap citra yang uji coba merupakan citra pada dengan nilai ketetangaan terbaik dari data *training*, dimana *rank* 1 merupakan hasil *rank* dengan label kelas terbaik, *rank* 2 merupakan label kelas terbaik kedua, hingga *rank* terakhir yang merupakan *rank* label kelas terendah. Hasil *rank* ditentukan dari jarak terdekat antar *keypoint* dari data *testing* ke semua data *training*, sehingga urutan *rank* tersebut merupakan kumpulan label kelas terbaik diantara citra data *training* lainnya. Pengujian ini juga akan menampilkan citra dari semua *rank* tersebut. Hasil pengujian 10 *rank* terbaik dari suatu citra dilihat pada gambar 4.8.



(a) Input



(b) Rank output

**Gambar 4.8** (a) input (b) output hasil 10 rank terbaik untuk penentuan label kelas

Gambar 4.8 menunjukkan 10 rank terbaik untuk penentuan label kelas dari suatu citra ikan. Ikan yang ditampilkan pada rank 1 hingga rank 5, dan rank 8 merupakan jenis ikan yang serupa. Sedangkan pada rank 6, 7, dan 9 merupakan jenis ikan yang berbeda namun memiliki bentuk fisik yang hampir serupa, sedangkan pada rank akhir merupakan jenis ikan dengan jenis dan bentuk yang berbeda. Pemilihan label kelas tidak hanya dari rank yang tertinggi, namun juga berdasarkan kemunculan jenis ikan terbanyak atau mayoritas dari 10 rank tersebut. Rincian hasil klasifikasi tersebut dapat dilihat pada tabel 4.2.

Tabel 4.2 Rincian Hasil Klasifikasi Citra Ikan

Input	<b>cephalopholis_sonnerati5.jpg</b>
Rank 1	cephalopholis_sonnerati
Rank 2	cephalopholis_sonnerati
Rank 3	cephalopholis_sonnerati

Rank 4	<i>cephalopholis_sonnerati</i>
Rank 5	<i>cephalopholis_sonnerati</i>
Rank 6	<i>anampses_melanurus</i>
Rank 7	<i>anampses_melanurus</i>
Rank 8	<i>cephalopholis_sonnerati</i>
Rank 9	<i>anampses_melanurus</i>
Rank 10	<i>bothus_pantherinus</i>
Hasil Klasifikasi	<b><i>cephalopholis_sonnerati</i></b>
Tingkat Akurasi	<b>100%</b>
Waktu Komputasi Ekstraksi Fitur	<b>0.2147</b>
Waktu Komputasi Deteksi Ikan	<b>0.813317</b>

Dari 10 rank tersebut, dapat dilihat bahwa label kelas yang paling tepat merupakan *cephalopholis sonerrati*, karena merupakan rank tertinggi sekaligus label terbanyak dari urutan *rank*. Jenis ikan ini dapat dikenali karena ciri ikan yang sangat mencolok pada warna pola dan bentuk tubuhnya. Urutan *rank* yang dihasilkan tidak hanya menampilkan jenis ikan yang sama, tapi juga menampilkan label kelas dengan ciri pada pola dan bentuk badannya yang hampir menyerupai jenis ikan pada citra input. Waktu komputasi yang dibutuhkan dalam ekstraksi fitur pada pengujian ini hanya sebesar 0.2147 detik, dan waktu deteksi ikan hanya 0.813317 detik. Sehingga tidak membutuhkan waktu komputasi yang banyak dalam pengujian ini. Hasil akhir dari proses klasifikasi menggunakan metode ini dapat dilihat pada gambar 4.9. Gambar tersebut menunjukkan label kelas yang tepat berdasarkan *rank* yang ditemukan.



**Gambar 4.9** Label kelas hasil klasifikasi dari satu jenis ikan

Gambar 4.8 menampilkan label kelas yang ditentukan hasil rank tersebut telah dimasukkan dalam bentuk teks pada gambar input sebagai hasil output dari metode ini. Pada tabel 4.2, kita juga dapat melihat bahwa tingkat akurasi dalam mengenali satu jenis ikan ini hingga 100%. Hasil akurasi ditentukan dari kesesuaian antara nama file pada citra input dengan hasil label kelas yang tepat. Jika nama file citra tersebut sama dengan nama jenis ikan yang ditampilkan, maka hasil akurasi menunjukkan 100%, sedangkan jika nama jenis ikan tidak sama dengan nama file input, maka hasil akurasi menunjukkan nilai 0% yang artinya kesalahan dalam klasifikasi. Waktu komputasi yang dibutuhkan dalam proses ekstraksi fitur menggunakan metode ORB dalam pegujian satu citra ini sebesar 0.2147, sehingga waktu komputasi yang dibutuhkan dalam proses mengekstraksi fitur dari citra sangat rendah. Waktu yang dibutuhkan dalam mendeteksi ikan ini yaitu sebesar 0.813317. Sehingga, hasil akurasi dari metode ini sangat akurat dalam mengenali ikan di setiap objek, dan tidak membutuhkan waktu komputasi yang banyak dalam proses mengekstraksi maupun deteksi ikan.

#### **4.3.2.1 Pengujian Metode Klasifikasi Dengan Perbedaan Rotasi**

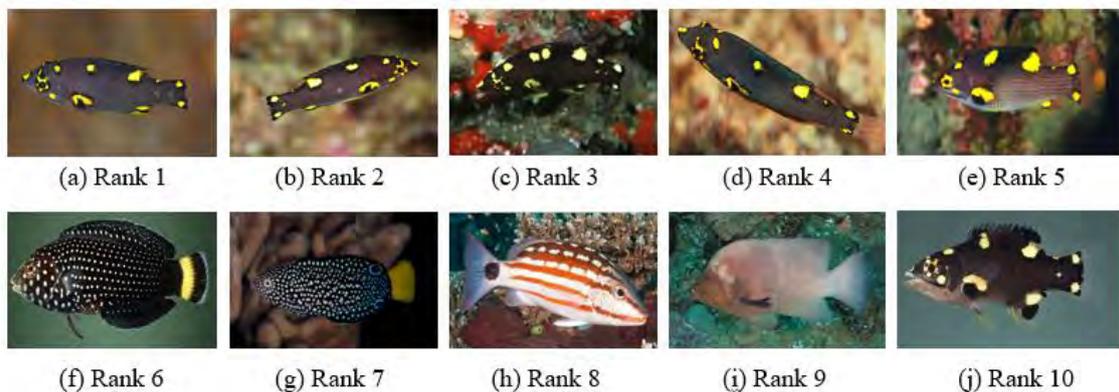
Perbedaan rotasi pada citra input juga sangat berpengaruh dalam klasifikasi objek. Karena ORB mempunyai kelebihan dalam rotasi invarian, maka fitur pada citra dengan rotasi yang berbeda tersebut juga dapat dengan mudah untuk dikenali. Pengujian rotasi ini dilakukan dengan menginputkan citra ikan dengan rotasi yang berbeda dari

data training. Pada pengujian ini, citra yang dimasukkan memiliki objek ikan dengan sudut rotasi  $80^\circ$ . Input pada pengujian ini dapat dilihat pada gambar 4.10.



**Gambar 4.10** Input pengujian rotasi

Dari gambar input tersebut, maka dilakukan proses dari ekstraksi fitur hingga klasifikasi. Hasil yang didapatkan ditentukan berupa 10 rank terbaik seperti pada gambar 4.10.



**Gambar 4.11** Hasil 10 rank ketetanggan terdekat pengujian rotasi

Hasil tersebut menunjukkan urutan 10 gambar dengan rank terbaik dari citra input. Walaupun data testing yang dimasukkan merupakan citra dengan sudut yang berbeda dengan data training, metode ini tetap mampu mengenali ikan tersebut. Terbukti dari hasil pengujian ini, dari 10 rank tersebut dapat dilihat bahwa ada 6 rank yang menampilkan jenis ikan yang sama seperti gambar input. Ikan tetap dapat dikenali karena memiliki ciri yang sama seperti pada kumpulan data training. Hasil akhir dari pengujian rotasi invarian dapat dilihat pada gambar 4.12.



**Gambar 4.12** Hasil pengujian pengenalan ikan dengan perbedaan rotasi

Label kelas ditentukan dari banyaknya label kelas yang sama dari rank yang ditampilkan pada gambar 4.11. Pada tahapan ini juga akan diuji beberapa citra dengan beberapa sudut rotasi yang berbeda, yaitu dengan sudut  $0^\circ$ ,  $25^\circ$ ,  $50^\circ$ ,  $90^\circ$ , dan  $210^\circ$ . Hasil pengujian rotasi tersebut dapat dilihat pada tabel 4.3.

Tabel 4.3 Hasil Pengujian klasifikasi dengan berbagai sudut rotasi

Rotasi	$0^\circ$	$25^\circ$	$50^\circ$	$80^\circ$	$210^\circ$
Input					
Akurasi	100%	100%	100%	100%	100%
Waktu Ekstraksi	0.285916	0.449418	0.323021	0.112861	0.184141
Waktu Deteksi	1.07366	1.29098	0.895546	0.766998	0.894646

Hasil dari pengujian ini membuktikan bahwa fitur yang dihasilkan oleh metode ORB dapat dikenali karena fitur yang dihasilkan tahan terhadap segala perubahan rotasi. Perbedaan rotasi pada ikan juga tidak mempengaruhi waktu yang dibutuhkan dalam komputasi dalam mengekstraksi fitur maupun deteksi jenis ikan.

#### 4.3.2.2 Pengujian Metode Klasifikasi Pada Keseluruhan Data Uji

Tahap ini dilakukan pengujian metode yang diusulkan dengan menguji keseluruhan data pada data uji dan data training. Masing-masing data uji dilakukan klasifikasi dengan mengurutkan 10 rank nama citra yang paling cocok dari semua citra data training. Hasil dari pengujian ini akan menampilkan info dari input data, 10 rank terbaik dari masing-masing citra uji, tingkat akurasi secara keseluruhan, dan keterangan rata-rata waktu yang dibutuhkan dalam melakukan ekstraksi fitur dan deteksi objek pada masing-masing data uji. Hasil dari pengujian ini dapat dilihat pada tabel 4.4.

Tabel 4.4 Hasil Klasifikasi Keseluruhan Data

Jenis Ikan	Jumlah Data		Hasil Klasifikasi		Tingkat Akurasi
	Testing	Training	Benar	Salah	
<i>Acanthistius cinctus</i>	3	8	3	0	100%
<i>Acreichthys tomentosus</i>	2	8	2	0	100%
<i>Aethaloperca rogae</i>	2	10	2	0	100%
<i>Aluterus monoceros</i>	2	9	2	0	100%
<i>Aluterus scriptus</i>	4	10	4	0	100%
<i>Amanses scopas</i>	6	10	6	0	100%
<i>Anampses Melanurus</i>	2	8	2	0	100%
<i>Bodianus anthioides</i>	2	8	2	0	100%
<i>Bodianus mesothorax</i>	3	8	3	0	100%
<i>Bothus pantherinus</i>	2	8	2	0	100%
<i>Cephalopholis sonnerati</i>	3	7	3	0	100%
<i>Halichoeres chrysus</i>	3	8	2	1	67.8%
<i>Lutjanus decussatus</i>	2	9	2	0	100%
<i>Lutjanus sebae</i>	2	7	2	0	100%
<i>Thalassoma hardwicke</i>	2	7	2	0	100%

TOTAL	40	125	39	1	97.5%
-------	----	-----	----	---	-------

Pada tabel 4.3, dapat dilihat bahwa hasil klasifikasi menggunakan metode yang diusulkan dapat menghasilkan tingkat akurasi hingga 97.5%. Hasil klasifikasi dari metode ini sangat dipengaruhi oleh citra pada data training ataupun data uji. Semakin banyak citra pada data training akan menghasilkan akurasi yang lebih baik, namun banyaknya data tersebut akan mengakibatkan proses ekstraksi atau deteksi menjadi lebih lama. Waktu yang dibutuhkan dalam metode ini dapat dilihat pada tabel 4.5.

Tabel 4.5 Waktu komputasi proses klasifikasi keseluruhan data

Waktu Komputasi Ekstraksi Fitur (detik)	
Total waktu Ekstraksi fitur dari keseluruhan data	11.5467
Rata-rata waktu ekstraksi fitur tiap citra	0.28657
Waktu Komputasi Deteksi Ikan (detik)	
Total waktu Deteksi dari keseluruhan data	39.5463
Rata-rata Deteksi tiap citra	0.97352

Seperti yang ditunjukkan pada tabel 4.5, waktu komputasi yang dibutuhkan dalam mengekstraksi fitur keseluruhan ikan adalah sebesar 11.5467 detik dengan rata-rata 0.28657 tiap citra, dan waktu yang dibutuhkan dalam mendeteksi ikan di keseluruhan data tersebut adalah 39.5463 detik dengan rata-rata 0.97352 di setiap citra. Waktu yang dibutuhkan dalam ekstraksi fitur atau mendeteksi ikan dipengaruhi oleh fitur yang ditemukan pada setiap citra. Ukuran dimensi pada citra juga sangat mempengaruhi dalam proses klasifikasi tersebut. Semakin kecil ukuran dimensi dari fitur yang dihasilkan, maka proses dalam klasifikasi akan lebih cepat.

### 4.3.3 Pengujian Dengan Metode Lain

Metode yang diusulkan dalam penelitian ini juga akan dibandingkan dengan beberapa metode lainnya. Metode yang akan dijadikan perbandingan, yaitu metode ORB-KNN, BRISK-PCA-KNN, dan BRISK-KNN. Data yang digunakan pada Pengujian yaitu semua dataset yang digunakan pada penelitian ini. Pengujian ini dilakukan dengan cara mengklasifikasi keseluruhan data secara langsung. Kemudian tingkat akurasi dan waktu komputasi yang dibutuhkan dalam klasifikasi citra akan diperbandingkan disetiap metode yang diuji.

#### 4.3.3.1 Pengujian Metode Klasifikasi Menggunakan ORB-KNN

Hampir sama seperti metode yang diusulkan, metode yang akan diuji coba yaitu menggunakan metode ORB-KNN tanpa melalui proses reduksi dimensi menggunakan PCA. Pengujian ini juga berguna untuk menguji pengaruh PCA dalam mengklasifikasi objek. Pengujian ini dilakukan dengan cara mengklasifikasi keseluruhan data secara langsung. Pengujian ini menghasilkan tingkat akurasi yang sama seperti metode yang diusulkan. Hasil dari pengujian ini dapat dilihat pada tabel 4.6.

Tabel 4.6 Hasil klasifikasi keseluruhan data menggunakan metode ORB-KNN

Tingkat Akurasi	
Tingkat Akurasi	<b>97.5 %</b>
Waktu Komputasi Ekstraksi Fitur (detik)	
Total waktu Ekstraksi fitur dari keseluruhan data	13.1104
Rata-rata waktu ekstraksi fitur tiap citra	0.32776
Waktu Komputasi Deteksi Ikan (detik)	
Total waktu Deteksi dari keseluruhan data	45.5157
Rata-rata Deteksi tiap citra	1.21289

Dapat dilihat dari tabel 4.6, hasil akurasi yang dihasilkan menggunakan ORB-KNN juga mencapai 97.5%, sama dengan tingkat akurasi yang dihasilkan oleh metode yang diusulkan yang menggunakan ORB-PCA dan KNN. Waktu yang dibutuhkan dalam proses ekstraksi fitur juga tidak jauh berbeda. Sedangkan waktu yang dibutuhkan dalam proses deteksi atau mengenali ikan, metode tanpa menggunakan PCA ini

membutuhkan waktu sebesar 45.5157 detik di keseluruhan citra atau dengan rata-rata 1.21289 detik disetiap citra, waktu yang dibutuhkan lebih lama dibandingkan dengan metode ORB-PCA dan KNN yang membutuhkan waktu deteksi dan mengenali ikan dalam waktu 30.1045 detik atau dengan rata-rata 0.752612 detik disetiap citra. Proses tanpa PCA membutuhkan waktu yang lebih lama dalam mengklasifikasi objek karena ukuran dimensi pada fitur yang dihasilkan oleh ORB yang berukuran besar. Ukuran dimensi dalam fitur juga sangat berpengaruh dalam proses kecepatan klasifikasi. Sehingga proses reduksi menggunakan PCA juga dibutuhkan dalam proses klasifikasi objek.

#### 4.3.3.2 Pengujian Metode Klasifikasi Menggunakan BRISK-PCA-KNN

Selain dilakukan pengujian dengan metode tanpa PCA, penelitian ini juga akan dibandingkan dengan metode ekstraksi fitur yang lain, yaitu menggunakan *Binary Robust Invariant Scalable Keypoints* (BRISK), yang ditambahkan metode *Principal Component Analysis* (PCA), dan *K-Nearest Neighbor* (KNN). Sama seperti ORB, BRISK juga dapat mengekstraksi fitur dengan cara mendeteksi dan mendeskripsikan keypoint. Hasil dari pengujian melalui metode ini dapat dilihat pada tabel 4.7

Tabel 4.7 Hasil klasifikasi menggunakan metode BRISK-PCA-KNN

Tingkat Akurasi	
Tingkat Akurasi	<b>80 %</b>
Waktu Komputasi Ekstraksi Fitur (detik)	
Total waktu Ekstraksi fitur dari keseluruhan data	3.03686
Rata-rata waktu ekstraksi fitur tiap citra	0.0759216
Waktu Komputasi Deteksi Ikan (detik)	
Total waktu Deteksi dari keseluruhan data	9.58764
Rata-rata Deteksi tiap citra	0.239691

Dari hasil pengujian tersebut, dapat dilihat bahwa waktu komputasi metode BRISK-PCA-KNN lebih cepat dibandingkan metode yang diusulkan, yaitu hanya sebesar 3.03686 detik untuk mengekstraksi fitur, dan 9.58764 detik untuk deteksi ikan di semua dataset. Namun tingkat akurasi yang dihasilkan dari BRISK-PCA-KNN ini hanya sebesar 80%. Metode yang diusulkan jauh lebih unggul dalam tingkat akurasi,

karena metode BRISK menghasilkan jumlah fitur yang lebih sedikit dibandingkan ORB saat mengekstraksi ciri pada citra. Rincian hasil klasifikasi keseluruhan data menggunakan BRISK-PCA-KNN ini dapat dilihat pada lampiran A.

#### 4.3.3.3 Pengujian Metode Klasifikasi Menggunakan BRISK- KNN

Pengujian tahap akhir dari penelitian ini yaitu pengujian menggunakan *Binary Robust Invariant Scalable Keypoints* (BRISK), kemudian langsung diproses menggunakan *K-Nearest Neighbor* (KNN), tanpa menggunakan PCA seperti pengujian sebelumnya. Hasil dari pengujian melalui metode ini dapat dilihat pada tabel 4.8.

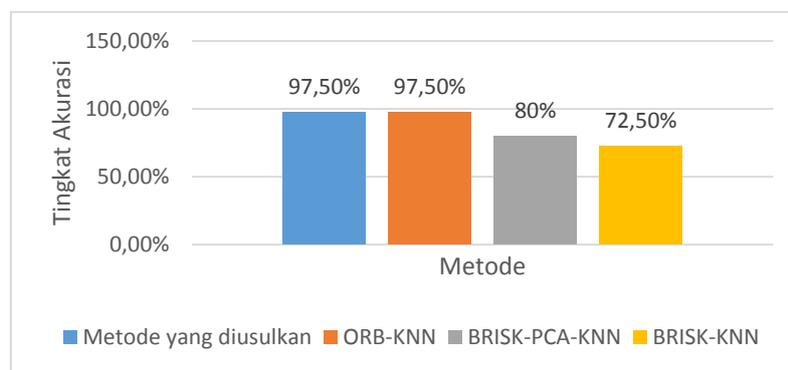
Tabel 4.8 Hasil klasifikasi menggunakan metode BRISK-KNN

Tingkat Akurasi	
Tingkat Akurasi	72.5 %
Waktu Ekstraksi Fitur (detik)	
Total waktu Ekstraksi fitur dari keseluruhan data	3.2614
Rata-rata waktu ekstraksi fitur tiap citra	0.081535
Waktu Deteksi Ikan (detik)	
Total waktu Deteksi dari keseluruhan data	10.1664
Rata-rata Deteksi tiap citra	0.254161

Dari hasil pengujian tersebut, metode ini menghasilkan akurasi yang lebih rendah dan membutuhkan waktu komputasi yang lebih banyak dibandingkan dengan metode BRISK-PCA-KNN. Tingkat akurasi yang dihasilkan dari metode ini hanya 72.5% dengan waktu komputasi ekstraksi fitur hingga 3.2614 detik dan waktu komputasi deteksi ikan sebanyak 10.1664 detik. Rincian hasil klasifikasi keseluruhan data menggunakan BRISK-PCA-KNN ini dapat dilihat pada lampiran B. Jika dibandingkan dengan ORB, tingkat akurasi dari metode ini juga masih unggul. Namun, waktu komputasi dari metode ini lebih sedikit dibandingkan ORB, karena fitur yang dihasilkan menggunakan metode ini tidak sebanyak ORB.

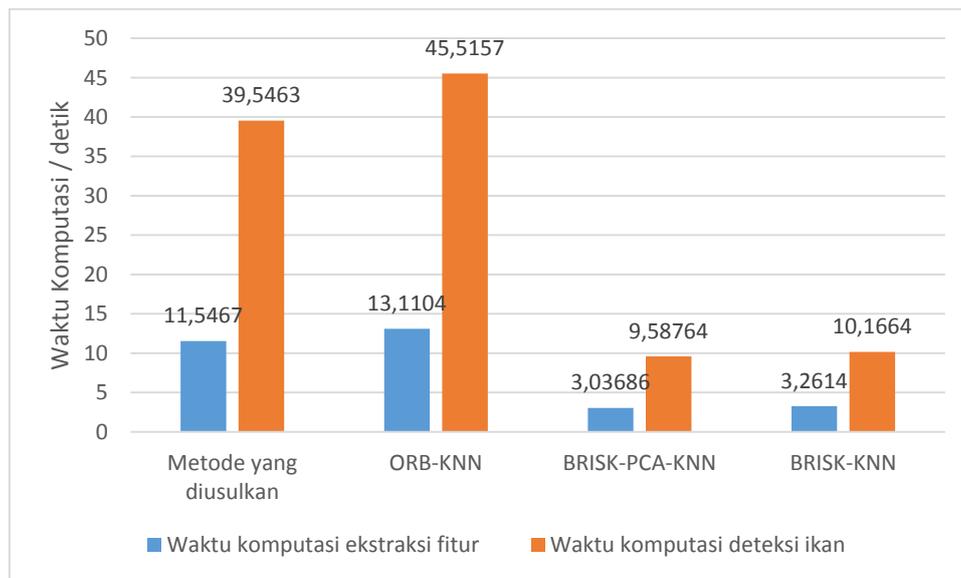
#### 4.4 Analisa Hasil Uji Coba

Berdasarkan hasil uji coba, setiap tahapan uji coba membuktikan bahwa metode yang diusulkan memperoleh hasil yang akurat dalam mengklasifikasi citra ikan. Selain itu, hasil klasifikasi citra ikan pada penelitian ini menghasilkan fitur yang dideskripsikan dengan baik melalui ORB, sehingga dapat mengenali ikan dengan rotasi yang invarian. Hal ini dapat dilihat pada diagram rata-rata akurasi setiap tahapan metode pada Gambar 4.13. Hal ini menunjukkan bahwa hasil setiap tahapan dari metode ini memiliki akurasi yang tinggi, khususnya performa algoritma ORB dalam mengekstraksi fitur dibandingkan tahapan lain yang menjadi pembandingnya.



**Gambar 4.13** Perbandingan akurasi metode klasifikasi ikan

Proses klasifikasi ikan menggunakan metode yang diusulkan dapat mencapai tingkat akurasi hingga 97.5%. Metode menggunakan ORB-KNN juga memiliki tingkat akurasi yang sama dengan metode yang diusulkan. Metode BRISK-PCA-KNN memiliki tingkat keberhasilan yang cukup tinggi, yaitu sebesar 80%. Sedangkan, metode BRISK-KNN memiliki tingkat keberhasilan terendah, yaitu sebesar 72.5%. Dalam hal ini, tahapan ekstraksi fitur terbaik adalah menggunakan ORB dalam tingkat akurasi yang dihasilkan. Selain itu, dari semua tahapan pengujian tersebut, juga dilakukan pengujian terhadap waktu komputasi yang dibutuhkan saat proses pengenalan ikan. Waktu komputasi yang diperbandingkan yaitu proses dalam mengekstraksi fitur, dan waktu komputasi dalam mendeteksi jenis ikan. Hasil perbandingan tersebut dapat dilihat pada diagram yang ditunjukkan pada gambar 4.14. Gambar 4.14 menunjukkan perbandingan antara waktu komputasi yang dibutuhkan oleh setiap metode dalam mengekstraksi fitur dan mendeteksi jenis ikan dari seluruh dataset. Semakin kecil waktu komputasi yang dibutuhkan, maka semakin cepat proses yang dilakukan.



**Gambar 4.14** Perbandingan waktu komputasi metode klasifikasi ikan

Metode tercepat dalam memproses klasifikasi ikan dalam pengujian ini adalah metode BRISK-PCA-KNN dengan waktu komputasi hanya 3.03686 detik untuk mengekstraksi fitur dan 9.58764 detik untuk mendeteksi jenis ikan. Disusul oleh metode BRISK-KNN yang memiliki waktu komputasi sebesar 3.2614 dan 10.1664 detik untuk mengekstraksi fitur dan mendeteksi jenis ikan dari seluruh dataset. Kemudian, metode yang diusulkan membutuhkan waktu yang sedikit lebih lama dalam mengklasifikasi ikan, dengan waktu komputasi masing-masing sebesar 11.5467 dan 39.5463 detik. Dan terakhir adalah metode ORB-KNN dengan waktu komputasi sebesar 13.1104 dan 45.5157 detik untuk mengekstraksi fitur dan mendeteksi jenis seluruh ikan pada dataset.

Dari hasil pengujian ini, dapat disimpulkan bahwa metode tanpa menggunakan PCA akan menambah beban waktu komputasi dari setiap metode. Hal ini dikarenakan banyaknya dimensi yang dihasilkan oleh setiap fitur akan mengakibatkan terbebannya proses klasifikasi menggunakan KNN. Selain membebani waktu komputasi, metode tanpa menggunakan PCA juga mempengaruhi nilai akurasi dalam klasifikasi. Hal ini terbukti pada pengujian metode BRISK-PCA-KNN yang dibandingkan dengan BRISK-KNN. Nilai akurasi BRISK-PCA-KNN lebih meningkat dibandingkan tanpa menggunakan PCA. Karena algoritma KNN mempunyai kelemahan yang rentan terhadap tingginya dimensionalitas. Semakin banyak dimensi, ruang yang bisa ditempati

*instance* semakin besar, sehingga semakin besar pula kemungkinan bahwa *nearest neighbour* dari suatu *instance*, sebetulnya sama sekali tidak “*near*“. Hal ini lah yang menyebabkan PCA sangat mempengaruhi nilai akurasi dan waktu komputasi. Sehingga algoritma PCA diperlukan untuk proses rekayasa fitur atau mengurangi dimensi dari fitur yang dihasilkan.

## **BAB 5**

### **KESIMPULAN DAN SARAN**

Berdasarkan uji coba dan analisis hasil pengujian pada metode klasifikasi ikan menggunakan ORB-PCA dan KNN, dapat diuraikan beberapa kesimpulan. Juga akan ditambahkan sedikit saran guna pengembangan penelitian kedepannya.

#### **5.1 Kesimpulan**

1. ORB dapat mendeteksi keypoint dan menghasilkan fitur yang invariant terhadap perubahan sudut rotasi pada citra ikan.
2. PCA sangat berpengaruh dalam proses klasifikasi menggunakan KNN, karena dapat mengurangi dimensionalitas dari hasil ekstraksi fitur ORB. Sehingga dapat meningkatkan akurasi, dan juga dapat mengurangi waktu komputasi dalam pengenalan ikan.
3. Dari hasil uji coba pengujian, metode yang diusulkan memiliki tingkat keberhasilan dalam mengklasifikasi seluruh sampel data ikan dengan tingkat akurasi sebesar 97.5% dengan nilai  $k=10$ .

#### **5.2 Saran**

Dalam penelitian ini penulis mengusulkan beberapa saran yang nantinya bisa diterapkan pada implementasi dan penelitian selanjutnya. Pertama, agar klasifikasi dapat dilakukan dengan lebih efisien, proses ini akan memerlukan tahapan *preprocessing* untuk mensegmentasi citra atau memisahkan antara objek dan background. Selain itu, Perlu dilakukan ujicoba menggunakan metode lain dalam melakukan klasifikasi ikan. Karena, algoritma KNN sangat bergantung pada nilai  $K$  dalam menentukan label kelas yang tepat. Sehingga diperlukan penelitian untuk menggunakan metode lain dalam mengklasifikasi ikan

*(halaman ini sengaja dikosongkan)*

## DAFTAR PUSTAKA

- A.Vinay, C., Akshay Kumar, Gaurav R. Shenoy, K. N. Balasubramanaya Murthya, S. Natarajan. (2015), "ORB-PCA Based Feature Extraction Technique for Face Recognition". *Procedia Computer Science*. vol. 58, Pages 614-621
- Akash P, D.R. Kasat, Sanjeev J, V M Thakare. (2014), "Performance Analysis of Various Feature Detector and Descriptor for Real-Time Video based Face Tracking", *International Journal of Computer Applications*. 2014 ; 93 :37-41.
- Endra, Utomo Budi. (2015), "Pengenalan Wajah Wanita Berkerudung Menggunakan Metode 2dpca Dan K-Nearest Neighbor". Skripsi, Fakultas Ilmu Komputer. Universitas Dian Nuswantoro.
- Ethan R, Vincent R, Kurt K (2011), "ORB: an efficient alternative to SIFT or SURF", *Computer Vision (ICCV)*. 2011; 2564-2571.
- I Wayan Agus Suryawibawa, I Ketut Gede Darma Putra, Ni Kadek Ayu Wirdiani. (2015), "Herbs Recognition Based on Android using OpenCV", *IJIGSP*. 7:1-7.
- Jiang, Y., Yang, X., Liu, S., Liao, L., (2013), "Application of Fishface Algorithm to Face Recognition System". *IEEE Conference Anthology*, Pages 1-4
- Kamencay. P, Hudec. R, Benco. M, Zachariasova. M. (2013), "Feature Extraction for Object Recognition using PCA-KNN with Application to Medical Image Analysis", *International Conference on Telecommunications and Signal Processing (TSP)*.
- Karami, E., Prasad, S., Shehata, M. (2015), "Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images", Newfoundland Electrical and Computer Engineering Conference.

- Kausari M., Rizal A., Adiwijaya.(2015), “Komputerisasi Iridologi Untuk Mendeteksi Kondisi Ginjal Menggunakan *Principal Component Analysis* (PCA) dan K-Nearest Neighbors (KNN)”, Laporan Tugas Akhir. Jurusan Teknik Telekomunikasi. Universitas Telkom.
- Kulkarni, A.V., J.S. Jagtap, and V. K. Harpale.(2013), “Object recognition with ORB and its Implementation on FPGA” *International Journal of Advanced Computer Research* 3, no. 3: 164-169.
- M. Mudrov , A. Procha ka. (2005), “Principal Component Analysis in image processing”, In: *Proceedings of MATLAB Technical Computing Conference*, Prague.
- Mulyono. T, Adi. K, Gernowo. R. (2012), “Sistem Pengenalan Wajah Dengan Metode Eigenface Dan Jaringan Syaraf Tiruan (JST)”, *Garuda*, Vol. 15, No. 1, Page 15 - 20
- Pawening, R. E., Arifin, A. Z. dan Yuniarti, A. (2016), “Ekstraksi Fitur Berdasarkan Deskriptor Bentuk dan Titik Salien Untuk Klasifikasi Citra Ikan Tuna”, *Jurnal Buana Informatika*, Vol.7, No.3
- Prashant A, Vijaykumar S K.,(2015), “Implementation Of High Performance Feature Extraction Method Using Oriented Fast And Rotated Brief Algorithm”, *International Journal of Research in Engineering and Technology*. 2015; 5 : 394-397.
- S. Leutenegger, M. Chli, and R.Y. Siegwart, (2011), “BRISK: Binary robust invariant scalable keypoints,” in 2011 IEEE International Conference on Computer Vision, pp. 2548–2555
- Santoso , Suharto Jati and Setiyono , Budi and Isnanto, R.Rizal (2011) “Pengenalan Jenis-Jenis Ikan Menggunakan Metode Analisis Komponen Utama”. Undergraduate thesis, Jurusan Teknik Elektro Fakultas Teknik Undip.

Spampinato, C, Giordano, D, Di Salvo, R, Chen-Burger, Y-H, Fisher, RB & Nadarajan, G. (2010), “Automatic fish classification for underwater species behavior understanding”. international workshop on Analysis and retrieval of tracked events and motion in imagery streams. ARTEMIS '10, ACM, New York, NY, USA, pp. 45-50.

Sumantara, I.G.L.T., Bayupati, I.P.A., Wirdiani, N.K.A.(2017), “Rancang Bangun Aplikasi Pengenalan Ukiran Bali dengan Metode ORB”. Merpati. Vol.5, Pages 51-56

T. Tuytelaars and K. Mikolajczyk. (2007), “Local Invariant Feature Detectors: A Survey,” *Found. Trends® Comput. Graph. Vis.*, vol. 3, no. 3, pp. 177–280.

*(halaman ini sengaja dikosongkan)*

## LAMPIRAN A

Jenis Ikan	Jumlah Data		Hasil Klasifikasi		Tingkat Akurasi
	Testing	Training	Benar	Salah	
<i>Acanthistius cinctus</i>	3	8	3	0	100%
<i>Acreichthys tomentosus</i>	2	8	2	0	100%
<i>Aethaloperca rogae</i>	2	10	2	0	100%
<i>Aluterus monoceros</i>	2	9	2	0	100%
<i>Aluterus scriptus</i>	4	10	3	1	75%
<i>Amanses scopas</i>	6	10	5	1	83%
<i>Anampses Melanurus</i>	2	8	2	0	100%
<i>Bodianus anthioides</i>	2	8	2	0	100%
<i>Bodianus mesothorax</i>	3	8	3	0	100%
<i>Bothus pantherinus</i>	2	8	2	0	100%
<i>Cephalopholis sonnerati</i>	3	7	3	0	100%
<i>Halichoeres chrysus</i>	3	8	2	1	67.8%
<i>Lutjanus decussatus</i>	2	9	2	0	100%
<i>Lutjanus sebae</i>	2	7	0	2	0%
<i>Thalassoma hardwicke</i>	2	7	0	2	0%
TOTAL	40	125	33	7	80%

## LAMPIRAN B

Jenis Ikan	Jumlah Data		Hasil Klasifikasi		Tingkat Akurasi
	Testing	Training	Benar	Salah	
<i>Acanthistius cinctus</i>	3	8	3	0	100%
<i>Acreichthys tomentosus</i>	2	8	2	0	100%
<i>Aethaloperca rogae</i>	2	10	2	0	100%
<i>Aluterus monoceros</i>	2	9	2	0	100%
<i>Aluterus scriptus</i>	4	10	3	1	75%
<i>Amanses scopas</i>	6	10	4	2	66.6%
<i>Anampses Melanurus</i>	2	8	2	0	100%
<i>Bodianus anthioides</i>	2	8	2	0	100%
<i>Bodianus mesothorax</i>	3	8	3	0	100%
<i>Bothus pantherinus</i>	2	8	2	0	100%
<i>Cephalopholis sonnerati</i>	3	7	2	1	67.8%
<i>Halichoeres chrysus</i>	3	8	0	3	0%
<i>Lutjanus decussatus</i>	2	9	2	0	100%
<i>Lutjanus sebae</i>	2	7	0	2	0%
<i>Thalassoma hardwicke</i>	2	7	0	2	0%
TOTAL	40	125	29	11	72.5%

## BIODATA PENULIS



Penulis tesis ini bernama Mirza Ramadhani, anak kedua dari tiga bersaudara, lahir pada tanggal 7 Maret 1992 dari pasangan H. Marhansyah dan Hj. Halimatus Sa'diah, S.Sos. Dilahirkan di Kota Banjarmasin, kota yang terkenal dengan sebutan “kota seribu sungai” yang berada di provinsi Kalimantan Selatan, Indonesia. Penulis menghabiskan masa kecilnya dengan bersekolah di kota kelahirannya sejak taman kanak-kanak hingga lulus Diploma 3. Penulis dinyatakan lulus dari SMA Negeri 3 Banjarmasin tahun 2010 dan melanjutkan studi Diploma 3 Teknik Informatika di Politeknik Negeri Banjarmasin dan lulus pada tahun 2013. Setelah lulus diploma 3, penulis mulai merantau dan langsung mengambil pendidikan ke luar kota untuk melanjutkan pendidikan sarjana S1 di Universitas Muhammadiyah Malang dengan mengambil jurusan yang sama, yaitu Teknik Informatika. Selain tekun dalam menjalani studi, penulis juga aktif dalam berwirausaha dibidang desain grafis. Penulis aktif dalam pembuatan jasa desain dan mengikuti kontes desain grafis dari dalam maupun luar negeri. Setelah lulus S1, penulis melanjutkan studi ke ITS jurusan Teknik Informatika. Selama kuliah di Teknik Informatika ITS, Penulis menyukai bidang minat Komputasional Cerdas visual (KCV). Kedepannya penulis berharap agar tetap terus berkarya dengan melakukan beberapa penelitian dan dapat membagi pengetahuannya kepada masyarakat. Penulis dapat dihubungi melalui email di [m2martdesign@gmail.com](mailto:m2martdesign@gmail.com).

*(halaman ini sengaja dikosongkan)*