

# Identifikasi Varietas Koi Berdasarkan Gambar Menggunakan Zero Parameter Simple Linear Iterative Clustering dan Support Vector Machine

Amadea Sapphira, Alexander Setiawan, Endang Setyati  
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra  
Jl. Siwalankerto 121-131 60236

Telp. (031)-2983455, Fax. (031)-8417658

E-mail: amadeasapphira.24@gmail.com, alexander@petra.ac.id, esetyati@gmail.com

## ABSTRAK

Terdapat 120 jenis ikan koi yang dikembangkan saat ini di seluruh dunia. Jenis-jenis ikan koi tersebut bergantung dengan corak warna dan bentuk yang mereka miliki. Tidak sedikit dari corak-corak itu memiliki kemiripan antara satu jenis dengan jenis yang lain. Contohnya, ikan koi jenis sanke dan showa bagi orang awam akan terlihat sama, karena keduanya memiliki corak warna yang sama, yaitu merah, hitam, dan putih. Padahal, koi jenis sanke lebih dominan ke warna merah dan putih dengan sedikit aksentuasi hitam, sedangkan showa dominan merah dan hitam, dengan aksentuasi putih.

Dalam penelitian ini, dilakukan pengujian pada dua metode *superpixel segmentation*, yaitu metode *Simple Linear Iterative Clustering* (SLIC) dan *Zero Parameter Simple Linear Iterative Clustering* (SLICO) untuk proses segmentasi objek untuk menghilangkan *background* pada gambar. Metode *Color Local Binary Pattern* digunakan untuk mendapatkan tekstur pada gambar melalui *colorspace* RGB, HSV, dan *Grayscale*. *Support Vector Machine* (SVM) digunakan untuk mengidentifikasi jenis ikan koi. Untuk menguji SVM, digunakan dua jenis kernel, yaitu kernel *linear* dan kernel *Radial Basis Function* (RBF).

Hasil dari penelitian ini adalah program dapat mengenali jenis ikan koi pada gambar. Hasil pengujian menunjukkan akurasi sebesar 36% pada *colorspace grayscale*, 50% pada *colorspace RGB*, dan 48% pada *colorspace HSV*.

**Kata Kunci:** *Color Local Binary Pattern, Support Vector Machine, Zero Parameter Simple Linear Iterative Clustering.*

## ABSTRACT

*There's currently 120 types of koi fish that has been bred around the world. The types of koi fish depends on the color patterns and shapes they have. There's alot of patterns that has similarity between one type with another. For example, sanke and showa koi fish will look similar from a non-expert's point of view, because both type has same color pattern, which is red, black and white. In actuality, sanke koi is dominantly red and white with slight black accent, while showa's dominant color is red and black, with white accent.*

*In this research, Zero Parameter Simple Linear Iterative Clustering (SLICO) method and Simple Linear Iterative Clustering (SLIC) will be tested and used to process the image segmentation process to eliminate the background of the image. Color Local Binary Pattern method is used to get the textures on images through the RGB, HSV, and grayscale colorspace. Support Vector Machine is used to identify types of koi fish. To test the SVM, two kind of*

*kernel is used, which is linear kernel and Radial Basis Function (RBF) kernel.*

*The results of this study are the program able to recognize types of koi from iamges. The test results show an accuracy of 36% in grayscale colorspace, 50% in RGB colorspace, and 48% in HSV colorspace.*

**Keywords:** *Color Local Binary Pattern, Support Vector Machine, Zero Parameter Simple Linear Iterative Clustering.*

## 1. PENDAHULUAN

Saat ini, terdapat 120 jenis ikan koi yang dikembangkan sejak tahun 1971 [6]. Jenis-jenis ikan koi tersebut bergantung dengan corak warna dan bentuk yang mereka miliki. Tidak sedikit dari corak-corak itu memiliki kemiripan antara satu jenis dengan jenis yang lain. Contohnya, ikan koi jenis sanke dan showa bagi orang awam akan terlihat sama, karena keduanya memiliki corak warna yang sama, yaitu merah, hitam, dan putih. Padahal, koi jenis sanke lebih dominan ke warna merah dan putih dengan sedikit aksentuasi hitam, sedangkan showa dominan merah dan hitam, dengan aksentuasi putih. Untuk mengantisipasi permasalahan tersebut, sebuah program untuk membantu identifikasi jenis ikan koi akan dapat membantu penggunaannya dalam mencari tahu jenis ikan koi yang mereka akan jual atau beli. Dengan menggunakan teknologi *Computer Vision*, hal ini mungkin untuk dicapai.

Sejauh ini, beberapa peneliti telah melakukan riset mengenai identifikasi jenis-jenis ikan. Salah satunya adalah penelitian yang melakukan identifikasi jenis ikan *herring*, *mackerel*, dan *blue whiting* dengan menggunakan CNN dan gambar sintetik, dengan hasil akurasi 94.1% [2]. Dalam penelitian mengenai identifikasi jenis ikan koi yang sudah pernah diadakan, peneliti menggunakan metode SLIC dan CNN. Walau hasil akhir akurasi tergolong bagus, yaitu 80%, namun dikarenakan parameter yang dibutuhkan tiap gambar berbeda-beda, maka peneliti harus memasukkan parameter secara manual [12]. Hal ini tentu akan cukup mempengaruhi hasil akhir. Apabila gambar yang baru dimasukkan membutuhkan parameter baru, hasil tidak akan seakurat sebelumnya karena parameternya tidak cocok.

Untuk membuat sebuah sistem identifikasi objek yang akurat, dapat menggunakan metode *superpixel segmentation*. *Superpixel* sendiri memiliki beberapa metode, dimana metode tersebut akan mengelompokkan *pixel-pixel* yang ada dalam gambar menjadi sekelompok *pixel* dengan karakteristik yang sama. *Simple Linear Iterative Clustering* (SLIC) merupakan salah satu metode *superpixel* yang dapat digunakan. Kekurangan dari metode ini adalah sulitnya menentukan parameter yang dibutuhkan tiap gambar, terlebih jika gambar yang digunakan memiliki kualitas and

tekstur yang berbeda. [1]. Untuk mengatasi kekurangan dari metode SLIC yang sudah disebutkan, dalam penelitian ini akan digunakan *Zero Parameter Simple Linear Iterative Clustering (SLICO) Superpixel* untuk segmentasi. Setelah gambar berhasil disegmentasi, pengenalan jenis ikan koi dapat dilakukan dengan *Local Binary Pattern (LBP)* dan *Support Vector Machine (SVM)*.

SVM adalah metode yang sering digunakan dalam hal klasifikasi data, gambar, maupun teks. SVM merupakan model yang kompak, sehingga memori yang digunakan tergolong sedikit. Selain itu, dengan menggunakan kernel yang dapat mengubah input menjadi format yang dibutuhkan, sehingga SVM mampu beradaptasi dengan berbagai data. Terdapat berbagai jenis kernel, diantaranya adalah kernel *linear* dan RBF. Kernel *linear* merupakan kernel yang paling sederhana, yang sering dipakai ketika data yang dimasukkan mampu dipisahkan hanya dengan sebuah garis saja. Kernel ini juga sering dipakai ketika dalam sebuah *dataset* terdapat *feature* dengan jumlah besar. Kernel *Radial Basis Function (RBF)* merupakan kernel yang sering digunakan untuk klasifikasi, dimana batas-batas yang dimiliki dihipotesiskan memiliki bentuk kurva. Kernel tersebut memiliki parameter yang mudah di kalibrasikan sehingga mudah untuk digunakan.

Dalam penelitian ini metode SLIC dan *Zero Parameter Simple Linear Iterative Clustering (SLICO)* akan diuji dan dibandingkan. Setelah gambar berhasil disegmentasi, metode SVM digunakan untuk melakukan identifikasi pada jenis ikan. Identifikasi menggunakan SVM akan dibantu oleh kernel, dimana dalam penelitian ini akan membandingkan hasil akhir dari dua jenis kernel, yaitu kernel *linear* dan kernel RBF.

Tujuan dari penelitian ini adalah untuk membuat sebuah program yang mampu mengidentifikasi jenis-jenis ikan koi berdasarkan corak dan bentuknya, dengan menggunakan gambar. Diharapkan dari penelitian ini, orang yang kurang paham mengenai jenis ikan koi dapat mengenali jenis ikan koi yang ada, juga agar dapat membantu pembeli maupun penjual ikan koi untuk mengecek atau *cross-check* kembali ikan yang akan diperjual-belikan.

## 2. DASAR TEORI

### 2.1 Koi

Ikan koi adalah ikan air tawar dan adalah jenis omnivore. Ikan ini beraktivitas selama musim panas, semi, dan gugur. Sementara pada musim dingin, ikan koi akan berhibernasi. Ikan koi sendiri adalah jenis ikan mas. Yang membuatnya berbeda dari ikan mas biasa adalah bentuk badannya yang seperti torpedo dengan umur bisa mencapai 30 tahun. Apabila dirawat dengan baik, ikan ini mampu mencapai umur lebih dari 100 tahun. Jenis-jenis ikan koi dipengaruhi tidak hanya oleh bentuk, tapi juga oleh coraknya. Ikan koi adalah ikan yang dirawat di kolam yang relatif sedang atau besar, bukan di akuarium seperti ikan hias lainnya. Oleh karena itu, corak tubuh bagian atas ikan koi lah yang menentukan jenis ikan koi tersebut. Jenis-jenis ini akan mempengaruhi kualitas dan harga jual-beli ikan koi tersebut. Selain itu, corak warna tiap jenis berbeda-beda, membuat tiap jenis ikan koi unik dan beragam.

### 2.2 SLICO

*Superpixel* merupakan suatu algoritma yang mengelompokkan *pixels* di daerah yang sama dan memiliki karakteristik yang sama. Dengan menggunakan *superpixel* sebagai metode segmentasi, kecepatan program dan kualitas hasil akhir akan meningkat drastis. *Superpixel* sendiri memiliki beberapa metode, salah satunya adalah *Simple Linear Iterative Clustering (SLIC)*.

SLIC dapat dianggap sebagai salah satu bentuk dari k-means, tetapi SLIC memiliki perbedaan dimana jumlah perhitungan jarak berkurang sebesar ukuran superpixel. Warna dan hubungan spasial kemudian akan digabungkan untuk memperbarui ukuran dan kepadatan *superpixel* tersebut [8]. SLIC menggunakan parameter yang sama untuk semua *superpixel* dalam gambar.

Penelitian ini menggunakan salah satu jenis SLIC, yaitu *Zero Parameter Simple Linear Iterative Clustering (SLICO)*, dimana SLICO tidak mengharuskan penggunaannya untuk menetapkan parameter untuk tiap *superpixel*-nya. Dengan begitu, hasil dari segmentasi akan dapat menjadi lebih halus dari jika menggunakan SLIC. Untuk mendapatkan ukuran yang tepat untuk tiap *superpixel* dalam sebuah gambar, perlu dicari pusat *superpixel* yang dapat dicari dengan algoritma pada Tabel 1.

Tabel 1. Algoritma SLIC[1]

N	Jumlah piksel pada <i>input</i> gambar
K	Jumlah <i>superpixel</i> yang digunakan untuk segmentasi pada <i>input</i>
N/K	Perkiraan ukuran tiap <i>superpixel</i>
$S = \sqrt{N/K}$	Untuk tiap <i>superpixel</i> berukuran kira-kira sama, akan ada pusat <i>superpixel</i> pada tiap <i>interval grid</i> S.

Algoritma secara lengkap dapat dilihat pada Gambar 1.

Algorithm 1 Efficient superpixel segmentation	
1:	Initialize cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ by sampling pixels at regular grid steps $S$ .
2:	Perturb cluster centers in an $n \times n$ neighborhood, to the lowest gradient position.
3:	<b>repeat</b>
4:	<b>for</b> each cluster center $C_k$ <b>do</b>
5:	Assign the best matching pixels from a $2S \times 2S$ square neighborhood around the cluster center according to the distance measure (Eq. 1).
6:	<b>end for</b>
7:	Compute new cluster centers and residual error $E$ {L1 distance between previous centers and recomputed centers}
8:	<b>until</b> $E \leq$ threshold
9:	Enforce connectivity.

Gambar 1. Algoritma SLIC [1]

### 2.3 Image Segmentation

Segmentasi mengacu pada proses mempartisi gambar menjadi beberapa segmen, yang bertujuan untuk menyederhanakan dan mengubah representasi gambar menjadi lebih bermakna. Segmentasi biasa digunakan untuk menemukan objek dan batas gambar [13]. Hasil segmentasi gambar adalah seperangkat segmen yang secara kolektif mencakup seluruh gambar, atau satu set kontur diekstraksi dari gambar.

#### 2.3.1 Gaussian Blur

*Gaussian Blur* adalah hasil dari mengaburkan suatu gambar menggunakan fungsi Gaussian. Fungsi ini biasa digunakan untuk mengurangi *noise* dan detail, yang akan digunakan pada tahap *pre-processing* untuk meningkatkan struktur gambar [5].

#### 2.3.2 Region Adjacency Graph

*Region Adjacency Graph (RAG)* adalah metode yang mampu memberi gambaran mengenai hubungan antara tiap region yang didapat dari segmentasi [4]. Metode segmentasi ini biasa digunakan untuk segmentasi gambar berwarna [14]. Untuk mendapatkan representasi berbasis graf dari sebuah data, dapat dengan menggunakan *mapping function* sebagai berikut:

$$LV: V \rightarrow R^d \quad (1)$$

$$L_E : E = (n_i, n_j) \rightarrow R^2 \quad (2)$$

### 2.3.3 Graph Cut

*Graph cut* adalah teknik segmentasi berbasis grafik efisien yang dapat berfungsi sebagai alat minimalisasi energi yang kuat yang menghasilkan solusi optimal secara global [14]. Batas dari pelabelan didefinisikan sebagai *edge set* dalam persamaan 3/

$$\partial L = \{e_{v,w} \in E \mid L(v) \neq L(w)\} \quad (3)$$

Dimana L merupakan batas dari pelabelan simpul. Hubungan antara pelabelan dan pemotongan dirangkum dalam dua pernyataan, yaitu untuk setiap graf  $G = (V, E)$  dan himpunan *edge*  $S \subseteq E$ , maka ada sebuah *vertex labeling* L dari graf G, sehingga S adalah  $\partial L$ , dimana S adalah potongan pada graf G [9].

## 2.4 Local Binary Pattern

*Local Binary Pattern* adalah sebuah operator tekstur yang simple dan efisien, dimana tiap *pixel* dari sebuah gambar akan diberi label dan menganggap hasilnya sebagai angka biner. LBP sangat populer dalam pemakaiannya, karena dengan menggunakan LBP, analisa gambar secara *real-time* dapat dilakukan. Proses LBP menggunakan matriks 3x3 yang akan diubah menjadi angka biner yang kemudian akan di konversi menjadi angka desimal dan disimpan dalam LBP 2D *array* **Error! Reference source not found.**

## 2.5 Color Local Binary Pattern

*Color Local Binary Pattern* (Color LBP) merupakan adaptasi dari metode *Local Binary Pattern* (LBP). Cara pengambilan descriptor untuk Color LBP sama dengan metode LBP biasanya, namun ditambahkan dengan warna pada prosesnya. Color LBP merupakan metode LBP yang diperuntukkan gambar berwarna, dan digunakan sebagai deskriptor untuk warna. Hasil dari Local Binary Pattern adalah histogram warna yang merupakan deskriptor efektif dari gambar berwarna **Error! Reference source not found.** Terdapat berbagai macam *color space* yang dapat digunakan dalam melakukan pengambilan features untuk Color LBP yaitu, RGB, HSV dan YIQ. RGB merupakan *color space* yang paling sering digunakan dalam gambar berwarna. HSV adalah *color space* yang mewakili sistem penglihatan manusia. Sedangkan YIQ mewakili *luminance*, *chrominance blue* dan *chrominance red* [3].

## 2.6 Support Vector Machine

*Support Vector Machine* (SVM) adalah model pembelajaran dengan algoritma yang menganalisa data untuk klasifikasi, identifikasi, dan analisa. Metode ini mampu meminimalisasi kesalahan klasifikasi empiris dan memaksimalkan margin geometris, sehingga SVM dikenal sebagai *maximum margin classifier* [16]. Pembuatan model SVM dibantu oleh kernel yang digunakan untuk mengubah input menjadi format yang dibutuhkan. Terdapat beberapa kernel yang dapat digunakan, salah satunya adalah kernel *linear* [10]. Kernel *linear* adalah kernel yang paling sederhana yang seringkali setara dengan non-kernel. Kernel ini sering dipakai ketika data yang dimasukkan mampu dipisahkan hanya dengan sebuah garis saja, atau ketika dalam sebuah *dataset* terdapat *feature* dengan jumlah besar. Persamaan 6 merupakan persamaan SVM dengan menggunakan kernel *linear*.

$$linear : K(x_i, x_j) = x_i^T x_j. \quad (6)$$

Selain kernel *linear*, terdapat juga kernel *polynomial* dan *Radial Basis Function* (RBF). *Polynomial* kernel baik digunakan untuk *training* data yang dinormalisasi, sementara RBF dapat membuat batas-batas keputusan yang kompleks. Kernel RBF memiliki *hyperparameter* yang lebih sedikit daripada polinomial. Kernel ini juga mampu memetakan sampel ke ruang dimensional yang lebih tinggi secara non-linear. Kernel tersebut juga mampu menangani kasus dimana relasi antara label dan atribut adalah nonlinear [6]. Persamaan 4 merupakan persamaan SVM menggunakan kernel *polynomial*. Persamaan 5 merupakan persamaan SVM menggunakan kernel RBF.

$$polynomial : K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0 \quad (4)$$

$$rbf : K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \quad (5)$$

Dalam penggunaannya, tiap kernel membutuhkan parameter. Untuk mendapatkan parameter terbaik yang dapat menghasilkan nilai akurasi yang baik, digunakan metode *Grid Search* untuk mencari set parameter terbaik. Set parameter tersebut kemudian akan digunakan untuk mendapatkan SVM model dengan cara melakukan *training* pada dataset yang sudah disiapkan. Model tersebut kemudian akan digunakan pada *testing* untuk mendapat nilai generalisasi akurasi nya.

## 3. ANALISA DAN DESAIN SISTEM

### 3.1 Dataset

Data yang digunakan dalam penelitian ini merupakan citra digital. Data berjumlah 1500 gambar dalam bentuk JPG maupun PNG. Data ikan koi yang digunakan awalnya akan diambil dari peternakan ikan koi "IWAKKOI" yang terletak di Malang dan "NISHIKIGOI" di yang terletak di Kediri, tapi karena adanya pembatalan jadwal pergi juga terhambat oleh epidemi, akhirnya data dikumpulkan melalui internet, yaitu melalui [www.cuttlebrookkoifarm.co.uk](http://www.cuttlebrookkoifarm.co.uk), [www.queni-koi.co.uk](http://www.queni-koi.co.uk), [www.kodamakoifarm.com](http://www.kodamakoifarm.com), dan [www.cascadekoi.co.uk](http://www.cascadekoi.co.uk), juga tambahan gambar yang dikirim via *whatsapp* oleh pemilik ternak koi "NISHIKIGOI". Pencarian gambar tiap jenis dibantu oleh pemilik ternak koi "NISHIKIGOI" yang adalah pakar dalam hal jenis ikan koi. Dataset yang dikumpulkan memiliki *background* berwarna biru yang kontras dengan tubuh ikan koi yang akan dideteksi dan merupakan gambar dari berbagai jenis ikan koi. Penempatan objek diusahakan ditengah agar seluruh tubuh ikan koi dapat terlihat. Sudut pengambilan gambar diambil dari atas, karena identifikasi ikan koi selalu melalui corak ikan koi ketika dilihat dari atas.

#### 3.1.1 Dataset Training

*Dataset Training* merupakan data yang dikumpulkan untuk proses *training* pada model SVM. Digunakan sebanyak 1200 gambar yang sudah dikumpulkan, dengan tiap jenis berjumlah 80 gambar.

#### 3.1.2 Dataset Testing

*Dataset testing* merupakan dataset yang akan digunakan untuk mengecek tingkat keakurasian dari program yang telah dibuat. Dataset ini merupakan kumpulan gambar yang belum digunakan sebagai dataset *training*. Untuk tiap jenisnya, disiapkan 20 gambar ikan koi.

## 3.2 Analisa Sistem

### 3.2.1 Image Segmentation

Dalam proses ini, input gambar akan diberi gaussian blur sebelum kemudian dilakukan proses SLICO dan SLIC sebagai pembanding. Hasil dari proses SLICO dan SLIC kemudian akan digunakan untuk membuat graf. Dengan menggunakan graf tersebut, program dapat melakukan *graph cut* pada gambar. Hasil dari *graph cut* akan digunakan untuk mengganti background pada input gambar. Gambar yang sudah diganti backgroundnya kemudian akan diberlakukan *bounding box*, dimana program akan memotong gambar hingga gambar hanya mencakup data yang diperlukan saja. Setelah itu gambar akan *downscaling* agar memiliki resolusi yang sama dengan gambar-gambar lainnya.

### 3.2.2 Color Local Binary Pattern

*Image* yang didapatkan dari proses sebelumnya dideskripsikan dengan metode LBP yang diaplikasikan pada *channel* warna yang terdapat pada gambar. Fungsi untuk menjalankan LBP membutuhkan *numpoints* dan radius dimana keduanya akan digunakan untuk mengambil berapa banyak *pixel* yang akan di deskripsi dengan radius yang diinginkan. Fungsi LBP akan memberi hasil berupa *feature vector*. Setelah melakukan perhitungan histogram, histogram akan dinormalisasi.

### 3.2.3 Image Identification menggunakan Support Vector Machine

Dalam melakukan identifikasi pada gambar dengan menggunakan hasil LBP, histogram akan dihitung berdasarkan *similarity*nya. Proses ini dapat dilakukan dengan menggunakan metode *Support Vector Machine*. Metode ini berguna untuk mempelajari perbedaan berbagai kelas tekstur. Metode yang digunakan adalah *Linear SVM* dan *RBF SVM*. Pertama *training* data akan dilakukan. *Training* diperlukan agar program dapat mengetahui perbedaan tekstur yang ada pada kelas-kelas yang berbeda. *Training* akan dilakukan pada jenis *colorspace* RGB, HSV dan *grayscale*. Dataset yang sudah disegmentasi dan dideskripsikan dengan menggunakan LBP akan *fit* dengan menggunakan parameter yang sudah disiapkan. Fungsi yang digunakan akan mengkalkulasi keakuratan pada data. Setelah fungsi selesai kalkulasi, model SVM akan disimpan dalam bentuk *joblib*.

Proses klasifikasi gambar menggunakan SVM dimulai setelah program melakukan proses *loading* data yang sudah dilakukan *training* ke SVM. Apabila data memuat *colorspace* lebih dari 1 *channel*, maka *channel* tersebut perlu digabung menjadi *array* dua dimensi dan dikecilkan menjadi *array* satu dimensi. Hal ini dikarenakan *linear SVM* hanya menerima *array* 1 dimensi saja. Dari proses ini akan didapatkan hasil berupa *class* dari jenis ikan koi yang sudah dilakukan prediksi.

## 4. HASIL EKSPERIMEN

### 4.1 Pengujian Terhadap Hasil Training

*Training* dilakukan dengan menggunakan hasil segmentasi SLICO dan SLIC dan deskripsi pada gambar menggunakan LBP. Hasil deskripsi kemudian digunakan untuk melakukan *training* pada SVM. Nilai parameter LBP yang diuji adalah *numPoint* 24 dengan radius 1 hingga 5. Parameter yang digunakan untuk SLICO adalah *region size* 20, *minimum element size* 10, dan *number of iterations* 500 dan 300. Untuk SLIC, ditambahkan parameter *ruler* sebesar 10.0. Data diuji dalam dua kernel SVM, yaitu *linear* dan *RBF*. Parameter C yang akan diuji adalah 1000. Parameter untuk *RBF* yang akan diuji adalah 1 dan 0.5. Dalam tabel, variabel A mewakili

metode SLICO dengan parameter *region\_size* 20, *min\_element\_size* 10, dan *num\_iteration* 500. Variabel B mewakili metode SLICO dengan parameter *region\_size* 20, *min\_element\_size* 10 dan *num\_iteration* 300. Variabel C mewakili metode SLIC dengan parameter *region\_size* 20, *min\_element\_size* 10 dan *num\_iteration* 500, dan variabel D mewakili metode SLIC dengan parameter *region\_size* 20, *min\_element\_size* 10, dan *num\_iteration* 300. Selanjutnya, *numPoint* akan disingkat menjadi NP dan radius disingkat menjadi R.

### 4.1.1 Training pada Grayscale

Tabel 2. Hasil Testing pada Linear SVM dengan Cost 1000

	NP	R								
	24	1	24	2	24	3	24	4	24	5
A	26%		31%		28%		32%		32%	
B	24%		34%		27%		35%		34%	
C	25%		23%		20%		21%		20%	
D	23%		21%		20%		24%		20%	

Tabel 3. Hasil Testing pada RBF SVM dengan Cost 1000

1000		NP	R								
		24	1	24	2	24	3	24	4	24	5
1	A	21%		26%		27%		31%		30%	
	B	24%		25%		24%		33%		29%	
	C	24%		24%		24%		28%		24%	
	D	27%		21%		24%		27%		22%	
0.5	A	19%		24%		21%		27%		29%	
	B	23%		22%		20%		28%		25%	
	C	21%		21%		21%		23%		22%	
	D	23%		18%		23%		24%		21%	

Tabel 2 adalah hasil *testing* dengan menggunakan model SVM yang sudah di *train*. Kolom yang *dihighlight* adalah kolom yang memiliki nilai *overfit* terkecil dari tiap variabel. Dari hasil *testing* dapat disimpulkan bahwa untuk ujicoba dengan *colorspace grayscale* dan *linear SVM* dengan cost 1000, variabel D memiliki perbandingan terkecil antara akurasi *training* dan *testing*nya, yaitu sebesar 1. Namun variabel D memiliki nilai akurasi yang kecil dibandingkan dengan yang lain.

Tabel 3 adalah hasil *testing* dengan menggunakan model SVM yang sudah di *train*. Kolom yang *dihighlight* adalah kolom yang memiliki nilai *overfit* terkecil dari tiap variabel. Dari hasil *testing* dapat disimpulkan bahwa untuk ujicoba dengan *colorspace grayscale* dan *linear SVM* dengan cost 1000, variabel C memiliki akurasi terbaik, dimana hasil *training* sebesar 28% dan *testing* sebesar 28%. Variabel ini dipilih karena diantara hasil uji coba yang lain, variabel C dengan gamma 1 tidak *overfit* ataupun *underfit*, variabel lain tidak menghasilkan akurasi yang lebih baik daripada variabel D.

### 4.1.2 Training pada RGB

Tabel 4 adalah hasil *testing* dengan menggunakan model SVM yang sudah di *train*. Kolom yang *dihighlight* adalah kolom yang memiliki nilai *overfit* terkecil dari tiap variabel. Dari hasil *testing* dapat disimpulkan bahwa untuk ujicoba dengan *colorspace* RGB dan *linear SVM* dengan cost 1000, variabel B memiliki akurasi terbaik, dimana hasil *training* sebesar 53% dan *testing* sebesar 50%, dengan perbedaan 3%. Variabel ini dipilih karena diantara hasil uji coba yang lain, variabel B memiliki nilai *overfit* terkecil dibandingkan dengan hasil akurasi yang lainnya.

**Tabel 4. Hasil Testing pada Linear SVM dengan Cost 1000**

	NP	R								
	24	1	24	2	24	3	24	4	24	5
A	47%		51%		51%		53%		55%	
B	48%		48%		50%		47%		56%	
C	34%		34%		33%		31%		36%	
D	38%		30%		32%		32%		37%	

**Tabel 5. Hasil Testing pada RBF SVM dengan Cost 1000**

1000		NP	R								
		24	1	24	2	24	3	24	4	24	5
1	A	37%		37%		39%		44%		48%	
	B	39%		39%		39%		46%		46%	
	C	36%		37%		34%		33%		38%	
	D	36%		35%		35%		36%		39%	
0.5	A	31%		35%		35%		45%		46%	
	B	33%		34%		34%		42%		40%	
	C	35%		33%		34%		30%		35%	
	D	35%		36%		34%		32%		38%	

Tabel 5 adalah hasil *testing* dengan menggunakan model SVM yang sudah di *train*. Kolom yang *dihighlight* adalah kolom yang memiliki nilai *overfit* terkecil dari tiap variabel. Dari hasil *testing* dapat disimpulkan bahwa untuk ujicoba dengan *colorspace* RGB dan RBF SVM dengan cost 1000, variabel D dengan gamma 0.5 memiliki akurasi terbaik, dimana hasil *training* sebesar 41% dan *testing* sebesar 35%, dengan perbedaan sebesar 6. Variabel ini dipilih karena diantara hasil uji coba yang lain, variabel D dengan gamma 0.5 memiliki nilai *overfit* terkecil dibanding variabel lain.

#### 4.1.3 Training pada HSV

**Tabel 6. Hasil Testing pada Linear SVM dengan Cost 1000**

	NP	R								
	24	1	24	2	24	3	24	4	24	5
A	46%		37%		41%		44%		47%	
B	47%		39%		42%		44%		44%	
C	39%		40%		32%		34%		29%	
D	39%		38%		33%		32%		33%	

**Tabel 7. Hasil Testing pada RBF SVM dengan Cost 1000**

1000		NP	R								
		24	1	24	2	24	3	24	4	24	5
1	A	36%		39%		35%		39%		41%	
	B	37%		37%		38%		38%		36%	
	C	37%		36%		32%		36%		36%	
	D	36%		38%		31%		33%		36%	
0.5	A	37%		33%		30%		31%		37%	
	B	38%		33%		33%		39%		29%	
	C	37%		33%		31%		36%		33%	
	D	37%		35%		34%		35%		36%	

Tabel 6 adalah hasil *testing* dengan menggunakan model SVM yang sudah di *train*. Kolom yang *dihighlight* adalah kolom yang memiliki nilai *overfit* terkecil dari tiap variabel. Dari hasil *testing* dapat disimpulkan bahwa untuk ujicoba dengan *colorspace* HSV dan *linear* SVM dengan cost 1000, variabel C memiliki akurasi terbaik, dimana hasil *training* sebesar 36% dan *testing* sebesar 34%, dengan perbedaan 2%. Variabel ini dipilih karena diantara hasil uji coba yang lain, variabel C memiliki nilai *overfit* terkecil dibandingkan dengan hasil akurasi yang lainnya.

Tabel 7 adalah hasil *testing* dengan menggunakan model SVM yang sudah di *train*. Kolom yang *dihighlight* adalah kolom yang memiliki nilai *overfit* terkecil dari tiap variabel. Dari hasil *testing*

dapat disimpulkan bahwa untuk ujicoba dengan *colorspace* HSV dan RBF SVM dengan cost 1000, variabel B dengan gamma 0.5 memiliki nilai *overfit* terkecil yaitu 5, dengan hasil akurasi pada *training* sebesar 44 dan pada *testing* sebesar 39.

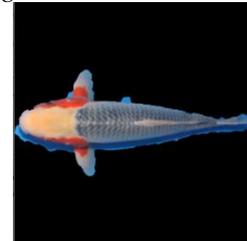
## 4.2 Pengujian Terhadap Rangkaian Program

Gambar 2 adalah gambar input yang digunakan untuk identifikasi jenis ikan koi.



**Gambar 2. Input**

### 4.2.1 Image Segmentation

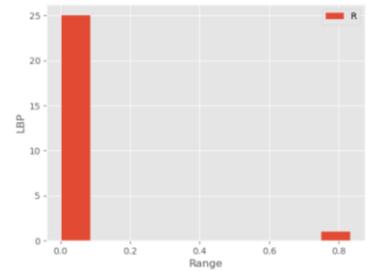


**Gambar 3. Hasil segmentasi**

Proses ini digunakan untuk menghilangkan background dan mengambil bentuk dan corak ikan koi saja. Proses ini dilakukan dengan Proses segmentasi ini memiliki tahapan-tahapan yaitu pemberian *Gaussian Blur*, aplikasi *SLICO* pada gambar, aplikasi *RAG* dan *Graph Cut* pada gambar, *replace background*, *bounding box object*, *crop object* dan *scaling*. Gambar 3 adalah gambar hasil segmentasi.

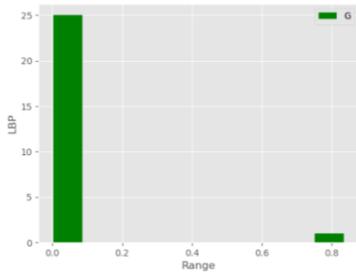
### 4.2.2 Local Binary Pattern

Gambar 4 adalah histogram untuk *color channel* R pada *colorspace* RGB yang didapatkan LBP.

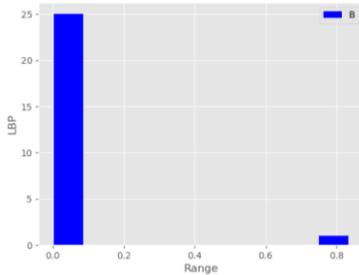


**Gambar 4. Hasil histogram untuk channel R pada colorspace RGB dari Local Binary Pattern**

Gambar 5 adalah histogram untuk *color channel* G pada *colorspace* RGB yang didapatkan dari LBP. Gambar 6 adalah histogram untuk *color channel* B pada *colorspace* RGB yang didapatkan dari LBP.

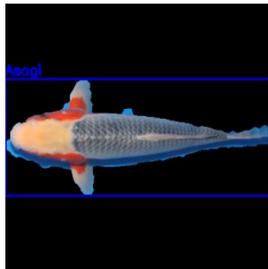


Gambar 5. Hasil histogram untuk channel G pada colorspace RGB dari Local Binary Pattern



Gambar 6. Hasil histogram untuk channel B pada colorspace RGB dari Local Binary Pattern

#### 4.2.3 Image Classification Menggunakan Support Vector Machine



Gambar 7. Hasil klasifikasi menggunakan SVM

Setelah proses deskripsi dengan menggunakan LBP selesai, histogram kemudian diklasifikasi dengan menggunakan *Support Vector Machine* (SVM). Dari proses klasifikasi akan didapatkan hasil berupa jenis ikan koi dari gambar input. Gambar 7 adalah contoh hasil prediksi dengan menggunakan SVM.

### 4.3 Perhitungan Akurasi untuk Masing-masing Jenis Ikan Koi

Pengujian pada Tabel 8 dilakukan dengan menggunakan *Local Binary Pattern* dengan numPoints sebesar 24 dan radius sebesar 3 dengan parameter SLICO yaitu region\_size 20, min\_element\_size 10, dan num\_iteration 300 dan *linear SVM* dengan parameter Cost 1000. Parameter tersebut dipilih karena memiliki jumlah akurasi tertinggi pada colorspace RGB sebesar 50% berdasarkan hasil perbandingan yang telah dilakukan sebelumnya.

Nilai yang tertulis pada tabel adalah nilai *precision* tiap jenis ikan koi. Nilai tersebut didapat dengan menggunakan fungsi `metrics.classification_report` dari *library sklearn*. Semakin tinggi nilai *precision*, semakin rendah kemungkinan terjadinya *false positive* pada saat pengujian. Pada colorspace grayscale, nilai tertinggi didapat oleh jenis ikan Asagi sebesar 52%, dan terendah

didapatkan oleh jenis ikan Hikarimoyomono sebesar 0%. Rata-rata dari nilai *precision* pada colorspace grayscale adalah 32.8%. Pada colorspace RGB, nilai tertinggi didapat oleh jenis ikan Shusui sebesar 88% dan terendah pada jenis ikan Matsuba sebesar 20%. Rata-rata nilai *precision* pada colorspace RGB adalah 52.9%. Pada colorspace HSV, nilai tertinggi didapat pada jenis ikan Shusui sebesar 75% dan terendah pada jenis ikan Tancho sebesar 24%. Rata-rata nilai *precision* pada colorspace HSV adalah 47%.

Tabel 8. Perbandingan Akurasi Terhadap Beberapa Colorspace

Jenis ikan koi	Grayscale (%)	RGB (%)	HSV (%)
Asagi	52	58	60
Bekko	22	70	36
Doitsu Hariwake	25	70	56
Goromo	20	65	33
Hikarimoyomono	0	33	46
Kawarimono	18	29	60
Kinginrin	45	45	42
Kohaku	50	63	58
Matsuba	33	20	31
Ogon	33	54	48
Sanke	50	71	54
Showa	47	38	58
Shusui	36	88	75
Tancho	23	54	24
Utsurimono	38	36	54

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan perancangan sistem, maka dapat disimpulkan bahwa:

- Dataset ikan koi yang dikumpulkan ada yang memiliki pantulan cahaya, bayangan atau riak air, menyebabkan beberapa gambar tidak dapat tersegmentasi dengan maksimal.
- Metode SLICO dapat menggantikan metode SLIC secara efisien, tanpa mengurangi waktu komputasi dimana kedua metode membutuhkan waktu 6 jam untuk memproses data *training* dan 4 jam untuk memproses data *testing*.
- Metode SLICO mampu memberi hasil akurasi yang lebih tinggi dibandingkan dengan metode SLIC. Pada colorspace RGB, SLICO mampu menghasilkan akurasi terbaik sebesar 50% sementara akurasi terbaik metode SLIC sebesar 35%. Pada colorspace HSV, SLICO mampu menghasilkan akurasi terbaik sebesar 39%, sementara akurasi terbaik metode SLIC sebesar 34%. Pada colorspace grayscale, SLICO mampu menghasilkan akurasi terbaik sebesar 34%, sementara akurasi terbaik metode SLIC sebesar 24%.
- Pemakaian parameter cost dan gamma pada RBF SVM dengan nilai besar mampu menaikkan nilai akurasi, namun rawan mengalami overfit.
- Metode *linear SVM* memberi hasil yang paling baik, dengan Cost sebesar 1000; namun hasil yang didapat masih mengalami *overfit* sebesar 3%.
- Pengujian menggunakan colorspace RGB dan *Local Binary Pattern* dengan numPoints sebesar 24 dan radius sebesar 3 dengan parameter SLICO yaitu region\_size 20, min\_element\_size 10, dan num\_iteration 300 dan *linear SVM* dengan parameter Cost 1000.

- Akurasi tertinggi untuk *color space Grayscale* dengan parameter yang sudah disebutkan pada poin sebelumnya didapatkan sebesar 36% .
- Akurasi tertinggi untuk *color space RGB* dengan parameter yang sudah disebutkan pada poin sebelumnya adalah 50%
- Akurasi tertinggi untuk *color space HSV* se dengan parameter yang sudah disebutkan pada poin sebelumnya besar 48%
- Jenis *color space* yang paling cocok digunakan dalam melakukan deskripsi menggunakan metode *Color Local Binary Pattern* adalah *color space RGB* dengan akurasi tertinggi sebesar 50%.

## 5.2 Saran

Berdasarkan hasil pengujian yang dilakukan pada sistem, maka disarankan:

- Improvisasi terhadap segmentasi gambar untuk memisahkan *background*.
- Mencari dataset yang lebih bersih, tidak banyak riak, bayangan ataupun pantulan cahaya karena dapat mempengaruhi segmentasi.
- Mencoba parameter lain untuk metode-metode yang digunakan.
- Menambahkan dataset sebagai salah satu cara mengatasi *overfit*.

## 6. DAFTAR PUSTAKA

- [1] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. 2012. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11), 2274–2282. doi: 10.1109/tpami.2012.120
- [2] Allken, V., Handegard, N. O., Rosen, S., Schreyeck, T., Mahiout, T., & Malde, K. 2018. Fish species identification using a convolutional neural network trained on synthetic data. *ICES Journal of Marine Science*, 76(1), 342–349. doi: 10.1093/icesjms/fsy147
- [3] Banerji, S., Verma, A., & Liu, C. 2012. *LBP and Color Descriptors for Image*.
- [4] Bodic, P. L., Locteau, H., Adam, S., Héroux, P., Lecourtier, Y., & Knippel, A. 2009. Symbol Detection Using Region Adjacency Graphs and Integer Linear Programming. *2009 10th International Conference on Document Analysis and Recognition*. doi: 10.1109/icdar.2009.2025
- [5] Gonzalez, R. C., & Woods, R. E. 2018. *Digital Image Processing* (4th, Illustr ed.). Pearson.
- [6] Hsu, C., Chang, C., & Lin, C. 2010. “A Practical Guide to Support Vector Classification” . Deptt of Computer Sci. National Taiwan Uni, Taipei, 106, Taiwan URI=<http://www.csie.ntu.edu.tw/~cjlin/2007>
- [7] Kock, S. D., & Gomelsky, B. 2015. Japanese Ornamental Koi Carp: Origin, Variation and Genetics. *Biology and Ecology of Carp*, 27–53. doi: 10.1201/b18547-4
- [8] Maghsoudi, O. H. 2017. Superpixel based segmentation and classification of polyps in wireless capsule endoscopy. *2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*. doi: 10.1109/spmb.2017.8257027
- [9] Malmberg, F. 2011. *Graph-based Methods for Interactive Image Segmentation*. Uppsala: Acta Universitatis Upsaliensis.
- [10] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. 2011. Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research*, 12.
- [11] Rosebrock, A. 2015. Local Binary Patterns with Python & OpenCV. Retrieved from pyimagesearch URI=<https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/>
- [12] Shandy, D. 2017. Pengenalan Varietas Ikan Koi Berdasarkan Foto Menggunakan Simple Linear Iterative Clustering Superpixel Segmentation dan Convolutional Neural Networks. Skripsi. Tidak Diterbitkan. Fakultas Teknik Informatika. Sekolah Tinggi Teknik Surabaya; Surabaya.
- [13] Shapiro, L., & Stockman, G. 2001. *Computer Vision* (illustrate). Prentice Hall.
- [14] Sharma, H., Alekseychuk, A., Leskovsky, P., Hellwich, O., Anand, R., Zerbe, N., & Hufnagl, P. 2012. Determining similarity in histological images using graph-theoretic description and matching methods for content-based image retrieval in medical diagnostics. *Diagnostic Pathology*, 7(1). doi: 10.1186/1746-1596-7-134
- [15] Singh, C., Walia, E., & Kaur, K. P. 2018. Color texture description with novel local binary patterns for effective image retrieval. *Pattern Recognition*, 76, 50–68. <https://doi.org/10.1016/j.patcog.2017.10.021>
- [16] Srivastava, Durgesh & Bhambhu, L.. 2010. Data classification using support vector machine. *Journal of Theoretical and Applied Information Technology*. 12. 1-7.
- [17] Zdimalova, Maria & Z., Kriv,á, & Bohumel, T.. 2015. Graph cuts in image processing. *APLIMAT 2015 - 14th Conference on Applied Mathematics, Proceedings*. 774-7