

# IMPLEMENTASI METODE *SUPPORT VECTOR MACHINE* PADA SISTEM PENGENALAN JEJAITAN

Ni Ketut Dewi Ari Jayanti<sup>1)</sup>, Kadek Dwi Pradnyani Novianti<sup>2)</sup>, I Wayan Sumalya<sup>3)</sup>

<sup>1), 2) 3)</sup> Sistem Informasi STMIK STIKOM Bali

Jl Raya Puputan No. 86 Renon Denpasar Bali

Email : [ari.jayanti@gmail.com](mailto:ari.jayanti@gmail.com)<sup>1)</sup>, [novianti@stikom-bali.ac.id](mailto:novianti@stikom-bali.ac.id)<sup>2)</sup>, [wmalya@gmail.com](mailto:wmalya@gmail.com)<sup>3)</sup>

## Abstrak

Jenis jejaitan untuk keperluan Upakara sangat beraneka ragam memiliki motif yang susah dibedakan antara yang satu dengan yang lain. Pada penelitian ini objek yang digunakan adalah gambar sampel jejaitan yang umum digunakan sehari-hari. Pengolahan citra digital digunakan untuk mengekstrak fitur dari objek yang diteliti, fitur yang diambil pada penelitian ini adalah 7 fitur HU moment invariant dan nilai bit yang mewakili setiap piksel pada gambar. Proses klasifikasi digunakan metode *Support Vector Machine multiclass* "one again one". Sistem ini dibangun berbasis desktop dengan menggunakan bahasa pemrograman c#, Accord.Net, dan Afforge.Net. Konsep perancangan yang digunakan untuk pengembangan sistem adalah perancangan terstruktur konteks diagram, Data Flow Diagram level 0, Data Flow Diagram level 1, dan Entity Relationship Diagram. Sistem ini mencapai akurasi 78% dengan 50 buah data training dan 50 buah data testing, dan mencapai 80% dengan 50 buah data training dan 20 buah data testing. Sehingga dapat disimpulkan bahwa metode *Support Vector Machine* memiliki kemampuan klasifikasi yang sangat baik.

**Kata kunci:** Pemodelan, retakan, ledakan, serious games, visualisasi.

## 1. Pendahuluan

Upakara merupakan salah satu sarana untuk memuja kebesaran Tuhan khususnya oleh umat Hindu sebagai tanda rasa syukur kita atas karunia Tuhan, jenis upakara atau persembahan yang dibuat oleh umat Hindu sangat beraneka ragam mengikuti budaya yang berkembang pada daerah masing-masing. Khususnya di Bali jenis upakara sangat beraneka ragam mulai dari yang Nista, Madya, dan Utama. Untuk kelengkapan upakara tersebut perlu adanya berbagai perlengkapan sebagai simbol-simbol filsafat kebesaran Tuhan salah satunya adalah jejaitan. Setiap upakara memiliki jejaitan yang sangat beraneka ragam dengan bentuk yang berbeda-beda, karena semakin tinggi tingkatan upakara akan memiliki jenis jejaitan yang semakin rumit yang hanya bisa dikerjakan oleh segelintir orang di Bali. Untuk upakara sehari-hari yang tergolong tingkat nista juga memerlukan jejaitan sebagai pelengkap,

demikian juga dengan hari-hari suci yang jatuh setiap 15 hari, 30 hari, dan seterusnya memerlukan jenis upakara yang berbeda sehingga memerlukan jejaitan yang berbeda pula.

Dengan perkembangan teknologi dan informasi, maka diperlukan adanya suatu perangkat lunak yang dapat membantu pengenalan jenis-jenis jejaitan berdasarkan image jejaitan yang digunakan sebagai data input. Sehingga masyarakat Bali khususnya umat Hindu dapat mengetahui secara lengkap jenis jejaitan, mengetahui fungsi dan makna dari jejaitan tersebut. Serta dapat membuat dan mengimplementasikan jejaitan tersebut dalam bentuk upakara sesuai dengan keperluan dan kebutuhan yadnya. Sebelumnya telah dilakukan banyak penelitian tentang artificial intelligent yang khusus membahas klasifikasi dan pengenalan pola suatu objek berdasarkan citra dari objek tersebut, adapun metode yang umum digunakan dan dapat menghasilkan akurasi yang sangat baik adalah *Support Vector Machine (SVM)*.

*Support Vector Machine (SVM)* dikenal sebagai teknik pembelajaran mesin (*machine learning*) paling mutakhir setelah pembelajaran mesin sebelumnya yang dikenal sebagai *Neural Network (NN)*. Baik SVM maupun NN tersebut telah berhasil digunakan dalam pengenalan pola. Pembelajaran dilakukan dengan menggunakan pasangan data input dan data output berupa sasaran yang diinginkan. Pembelajaran dengan cara ini disebut dengan pembelajaran terarah (*supervised learning*)[1]. Dari uraian tersebut maka dalam penelitian ini akan dibahas mengenai bagaimana mengembangkan suatu sistem informasi yang dapat digunakan sebagai klasifikasi dan pengenalan jenis jejaitan untuk keperluan upakara menggunakan pengolahan citra digital dan metode *Support vector Machine*. Sehingga diharapkan nanti masyarakat Bali dapat membuat jejaitan sebagai sarana upakara secara mandiri.

## 2. Pembahasan

### 2.1. Pengolahan Citra Digital

Pengolahan citra digital adalah pemrosesan citra, khususnya menggunakan komputer untuk mengubah suatu citra menjadi citra dengan format yang berbeda. Klasifikasi citra tidak dapat langsung dilakukan karena itu diperlukan proses-proses *preprocessing* seperti *grayscale*, *black and white*, *smoothing*, *morphology*, dan *edge detection* (deteksi tepi) dan

mendapatkan fitur sesuai dengan format yang diinginkan.

**2.2. Citra Grayscale**

Citra grayscale merupakan citra digital yang hanya memiliki nilai kanal pada setiap piksel nya. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas citra. *Grayscale* dapat dihitung dengan persamaan berikut [3]:

$$Gray = 0.299 * R + 0.587 * G + 0.114 * B \dots\dots(1)$$

Keterangan:

- R : Komponen red
- G : Komponen green
- B : Komponen blue

**2.3. Citra Biner**

Citra *biner* adalah citra digital yang hanya memiliki dua kemungkinan nilai *pixel* yaitu hitam dan putih. Citra *biner* juga disebut sebagai citra B&W (*black and white*) atau citra monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap piksel dari citra *biner*. Berikut persamaan untuk mengubah citra keabuan menjadi citra *biner* dengan nilai ambang T [4]:

$$(x, y) = \begin{cases} 0, f(x, y) < T \\ 1, f(x) \geq T \end{cases} \dots\dots(2)$$

Keterangan:

- (x) : nilai *pixel* di titik x
- T : *threshold*

**2.4. Smoothing**

Operasi pelembutan dilakukan dengan mengganti intensitas *pixel* dengan rata-rata nilai *pixel* tersebut dengan nilai *pixel* tetangganya. Secara sederhana operasi pelembutan citra didefinisikan dalam persamaan berikut [4]:

$$g(x, y) = f(x, y) * h(x, y) \dots\dots(3)$$

Keterangan:

- g = hasil pelembutan citra
- f = citra asli
- h = *mean filter*

**2.5. Deteksi Tepi**

Dasar dari deteksi tepi adalah Perubahan intensitas yang besar dalam jarak yang singkat dipandang sebagai fungsi yang memiliki kemiringan yang besar. Kemiringan fungsi biasanya dilakukan dengan menghitung turunan pertama (*gradient*). Berikut persamaan *gradient* dalam notasi *vector* [3]:

$$\nabla f = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \dots\dots(4)$$

Keputusan apakah suatu *pixel* merupakan tepi atau bukan tepi dinyatakan dengan operasi pengambangan berikut [4]:

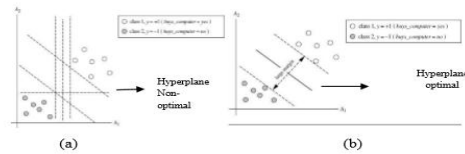
$$g(x, y) = \begin{cases} 1, jika M(x, y) \geq T \\ 0, lainnya \end{cases} \dots\dots(5)$$

Keterangan:

- (x,y) : nilai piksel di titik x,y
- (x) : magnitudo pada titik x,
- T : *threshold*
- α : arah tepi

**2.6. SVM (Support Vector Machine)**

*SVM* merupakan salah satu metode klasifikasi yang umum digunakan saat ini oleh banyak peneliti, karena memiliki kemampuan yang baik dalam banyak aplikasi. Ide dasar *SVM* adalah memaksimalkan batas *hyperplane*, yang diilustrasikan seperti gambar berikut:



**Gambar 1.** (a) *hyperplane non optimal* (b) *hyperplane optimal* [6].

Pada Gambar 1 (a) ada sejumlah pilihan *hyperplane* yang mungkin untuk set data, sedangkan Gambar 1 (b) merupakan *hyperplane* dengan margin yang paling maksimal. Meskipun sebenarnya pada Gambar 1 (a) bisa juga menggunakan *hyperplane* sembarang, tetapi *hyperplane* dengan margin yang maksimal akan memberikan generalisasi yang lebih baik pada metode klasifikasi. Konsep klasifikasi dengan *SVM* dapat dijelaskan secara sederhana sebagai usaha untuk mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas data pada input space. Data yang tergabung pada kelas -1 disimbolkan dengan bentuk lingkaran, sedangkan data pada kelas +1, disimbolkan dengan bentuk bujur sangkar [6].

Pada penelitian ini metode yang digunakan adalah metode *Support Vector Machine* untuk klasifikasi atau yang bisa disebut *SVC (Support Vector Classification)*. Klasifikasi adalah proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data dengan tujuan untuk memperkirakan kelas yang tidak diketahui dari suatu objek. Secara umum dalam proses klasifikasi memiliki dua proses yaitu:

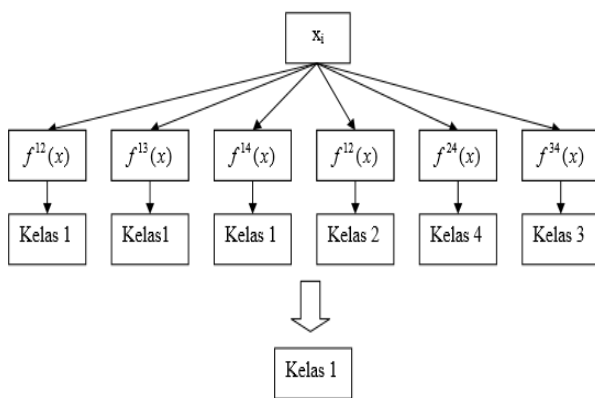
1. Proses *training*: pada proses *training* digunakan *training set* yang telah diketahui label-labelnya untuk membangun model atau fungsi.
2. Proses *testing*: untuk mengetahui keakuratan model atau fungsi yang akan dibangun pada proses *training*, maka digunakan data yang disebut dengan *testing set* untuk memprediksi label-labelnya.

Dalam *SVM*, untuk memisahkan data terhadap kelasnya, *SVM* akan membangun sebuah *hyperplane* (bidang pemisah). Sebuah *hyperplane* (bidang pemisah) yang baik, bukan hanya *hyperplane* yang bisa digunakan untuk memisahkan data, akan tetapi *hyperplane* yang

baik adalah *hyperplane* yang memiliki batasan (margin) yang paling besar. Pencarian bidang pemisah terbaik inilah yang menjadi inti dari *support* inti dari *support vector machine*. Akan tetapi, dalam proses pencarian *hyperplane* tersebut, akan muncul permasalahan baru yaitu sebuah formula yang sangat sulit untuk dipecahkan, yang disebut dengan permasalahan *Quadratic Programming*.

Pada penelitian ini kernel yang digunakan adalah kernel linier dan kernel RBF. Fungsi kernel yang direkomendasikan untuk diuji pertama kali adalah fungsi kernel RBF karena memiliki performansi yang sama dengan kernel linier pada parameter tertentu, memiliki perilaku seperti fungsi kernel *sigmoid* dengan parameter tertentu dan rentang nilainya kecil [0,1].

Metode *support vector machine* umumnya digunakan untuk klasifikasi dua kelas (SVM biner), namun seiring kebutuhan klasifikasi, SVM dapat digunakan untuk klasifikasi *multiclass* dengan menggabungkan beberapa SVM biner salah satu metode yang digunakan untuk klasifikasi *multiclass* adalah metode *one against one* Dengan menggunakan metode ini, dibangun  $\frac{k(k-1)}{2}$  buah model klasifikasi biner (k adalah jumlah kelas). Jika data x dimasukkan ke dalam fungsi hasil pelatihan  $f(x) = \sum_{i=1}^m a_i y_i K(x_i, y_i) + b$  dan hasilnya menyatakan x adalah kelas i, maka suara untuk kelas i ditambah satu. Kelas dari data x akan ditentukan dari jumlah suara terbanyak. Berikut adalah proses klasifikasi SVM *multiclass one against one*.



Gambar 2 Klasifikasi dengan SVM metode One-against-one [7].

Untuk melakukan proses klasifikasi maka perlu dilakukan pemodelan SVM untuk menguji data *training* dan data *testing*. Data *training* dihitung dengan menggunakan salah metode penyelesaian *training* data SVM yaitu *Sequential Minimal Optimization* (SMO). Hasil *training* adalah sebuah pembelajaran yang tersimpan di dalam sistem yang akan menjadi acuan bagi sistem untuk menentukan sebuah data tes jenis *jejaitan* baru yang diinputkan *user*. Penyelesaian dari langkah-langkah diatas dilakukan satu persatu seperti dibawah ini:

1. Menginisiasikan nilai awal  $a, c, \epsilon, \text{complexity}, \gamma$ .  
 $a = 0.5, c = 0, \epsilon = 0,01, \text{complexity} = 1,5, \gamma = 0,2$ .
2. Memasukan data uji. Data uji yang digunakan adalah data numerik yang didapatkan dari jumlah nilai 7 fitur *Hu moment invariant*.

Tabel 1 Contoh Data Training

No	Label	Fitur(X)	Kelas (y)
1	Canang	0.001006885	0
2	Ceper	0.001224034	1
3	Ceper	0.001352458	1
4	Canang	0.001096603	0

3. Menentukan *dot product* setiap data dengan memasukkan fungsi kernel (K). Fungsi kernel yang digunakan adalah fungsi kernel linier. Sebelumnya data di *transpose* karena menggunakan perkalian matriks  $A \times A^T$ .

Tabel 2 Transpose Data Uji

$X_1$	$X_2$	$X_3$	$X_4$
0.001006885	0.001224034	0.001352458	0.001096603

Pada metode kernel, data tidak direpresentasikan secara individual, melainkan lewat perbandingan antara sepasang data. Setiap data akan dibandingkan dengan dirinya dan data lainnya. Selengkapnya dapat dilihat pada tabel berikut ini:

Tabel 3 Perbandingan Data Uji Dan Transpose

$X_1$	$X_2$	$X_3$	$X_4$
$K[X_1, X_1]$	$K[X_1, X_2]$	$K[X_1, X_3]$	$K[X_1, X_4]$
$K[X_2, X_1]$	$K[X_2, X_2]$	$K[X_2, X_3]$	$K[X_2, X_4]$
$K[X_3, X_1]$	$K[X_3, X_2]$	$K[X_3, X_3]$	$K[X_3, X_4]$
$K[X_4, X_1]$	$K[X_4, X_2]$	$K[X_4, X_3]$	$K[X_4, X_4]$

Berikut adalah perhitungan dengan data uji dan data *transpose*:

$$K[X_1, X_1] = (0.001006885 * 0.001006885) = 1.01382E-06$$

Semua data dihitung dengan cara sama, baris x kolom sehingga menghasilkan nilai dot product seperti ditunjukkan oleh tabel dibawah:

Tabel 3 Hasil Perhitungan Kernel

$X_1$	$X_2$	$X_3$	$X_4$
1.01382E-06	1.23246E-06	1.36177E-06	1.10415E-06
1.23246E-06	1.49826E-06	1.65545E-06	1.34228E-06
1.36177E-06	1.36177E-06	1.82914E-06	1.48311E-06
1.10415E-06	1.34228E-06	1.48311E-06	1.20254E-06

4. Menghitung matriks dengan rumus berikut ini:

$$D_{ij} = y_i y_j (K(x_i, x_j) + \gamma^2)$$

Keterangan:

$D_{ij}$  = elemen matriks ke-ij.

$y_i$  = kelas data ke-i.

$y_j$  = kelas data ke-j.

$\lambda$  = batas teoritis yang akan diturunkan.

Contoh perhitungan untuk pasangan data  $X_1$ , dan  $X_1$ :

$$D_{ij} = (0)(0)( 1.01382E-06) + 0.5^2 = 0.25$$

Maka didapatkan hasil untuk semua data:

**Tabel 4 Hasil Perhitungan Matriks**

D X <sub>1</sub>	D X <sub>2</sub>	D X <sub>3</sub>	D X <sub>4</sub>
0.25	0.25	0.25	0.25
0.25	0.250001498	0.250003311	0.250004027
0.25	0.250002724	0.250007317	0.250008899
0.25	0.250004027	0.250008899	0.250010823

5. Mencari nilai error dengan rumus:

$$E_i = \sum_{j=1}^i a_j D_{ij}$$

Keterangan:

E<sub>i</sub> = nilai error data ke-i.

Maka didapatkan nilai error setiap data adalah:

$$E_i = (0.5 \cdot 0.25) + (0.5 \cdot 0.25) + (0.5 \cdot 0.25) + (0.5 \cdot 0.25) = 0,5$$

Nilai error selengkapnya dapat dilihat pada tabel:

**Tabel 5 Nilai Error**

Nilai Error
0.5
0.500004
0.500009
0.500012

6. Menghitung nilai delta alpha dengan rumus berikut:

$$\delta\alpha_i = \min\{\max[\gamma(1 - E_i), -a_i], C - a_i\}$$

Perhitungan delta alpha untuk data yang pertama adalah sebagai berikut:

$$\begin{aligned} \text{Delta alpha} &= \text{Min}(\max(0.5((1-0.5), -0.5), 1-0.5)) \\ &= \text{Min}(\max(0.5 \cdot 0.5, -0.5), 0.5) \\ &= \text{Min}(0.25, 0.5) \\ &= 0.25 \end{aligned}$$

Nilai delta alpha selengkapnya dapat dilihat pada tabel:

**Tabel 6 Nilai Delta Alpha**

Nilai delta alpha
0.25
0.249997791
0.249995265
0.249994063

7. Menghitung nilai  $\alpha$  baru dengan menggunakan rumus:

$$\text{new } \alpha_i = \alpha_i + \delta\alpha_i$$

Contoh perhitungan *new alpha*:

$$\text{Alpha} = 0.5 + 0.25$$

**Tabel 7 Nilai Alpha Baru**

Nilai delta alpha
0.75
0.749997791
0.749995265
0.749994063

8. Mencari nilai bias:

$$b = -\frac{1}{2} (\langle w \cdot x_1 \rangle + \langle w \cdot x_2 \rangle)$$

Terlebih dahulu dihitung nilai w:

Wi1 adalah bobot dot product data di kelas 0.

Wi2 adalah bobot dot product data dengan alpha terbesar di kelas lainnya.

$$\begin{aligned} W_{i1} &= (0 \times 0 \times 1.01382E-06) + (0 \times 1 \times 1.23246E-06) \\ &\quad + (0 \times 1 \times 1.36177E-06) + (0 \times 0 \times 1.10415E-06) = 0. \end{aligned}$$

$$\begin{aligned} W_{i2} &= (1 \times 0 \times 1.23246E-06) + (1 \times 1 \times 1.49826E-06) \\ &\quad + (1 \times 1 \times 1.65545E-06) + (1 \times 0 \times 1.34228E-06) = 8.83601E-06. \end{aligned}$$

$$\text{Maka nilai } b = -1/2 (w \cdot x + w \cdot x) = -1/2(0 + 8.83601E-06) = -4.418E06.$$

Sebenarnya perhitungan diatas belum bisa digunakan untuk fungsi keputusan karena iterasi masih harus diteruskan. Perhitungan dibawah ini hanya untuk contoh perhitungan dengan fungsi keputusan saja

9. Setelah mendapatkan nilai  $\alpha$ , w dan b maka dapat dilakukan pengujian dengan contoh data uji berikut:

**Tabel 8 Contoh Data Uji**

No	Label	Fitur(X)	Kelas (y)
1	Canang	0.001007315	0
2	Ceper	0.001224034	1

Dalam contoh perhitungan ini hanya akan dilakukan *training* dan *testing SVM* Biner yaitu dengan menggunakan dua buah kelas. Berikut ini adalah langkah-langkah pengujian nya:

Langkah pertama untuk menguji adalah menghitung dot product antara data uji dengan semua data latih dengan fungsi kernel.

$$K(x,y) = x \cdot y.$$

Dimana x adalah data uji dan y adalah semua data latih.

$$\begin{aligned} \text{Data ke-1 } K(x,y) &= (0.001007315 \times 0.001006885) + \\ &\quad (0.001007315 \times 0.001224034) + (0.001007315 \times 0.001352458) + \\ &\quad (0.001007315 \times 0.001096603) = 4.71421E-06 \end{aligned}$$

Demikian seterusnya sampai semua data latih dikalikan dengan semua data uji, sehingga hasil perkalian data uji dengan data dot product data latih seperti pada tabel berikut ini:

**Tabel 9 Hasil Perkalian Data Latih dan Data Uji**

Hasil data latih
4.71421E-06
5.72845E-06

Selanjutnya dilakukan perhitungan fungsi keputusan dengan rumus berikut:

$$f(x) = w \cdot x + b \text{ atau } f(x) = \sum_{i=1}^m a_i y_i K(x_i, y_i) + b$$

Data ke-1:

$$F(x) = \text{sign}((0 \times 0 \times 4.71421E-06) - 4.418E06 + (0 \times 1 \times 4.71421E-06) - 4.418E06 + (0 \times 1 \times$$



$$4.71421E-06) -4.418E06 + (0 \times 0 \times 4.71421E-06) -4.418E06.$$

$$= \text{sign}(0).$$

$$= 0$$

Jadi, data uji ke 1 diatas termasuk kelas 0 yang artinya sama dengan ekspektasi yang diharapkan.

### 2.7. Hasil

Sistem pengenalan jenis jejjaitan ini dibangun berbasis desktop menggunakan bahasa pemrograman C#, dan Database SQL Server. Serta memanfaatkan library machine learning Accord.Net dan Afforge.Net. Hasil dari sistem yang dikembangkan berfungsi untuk membantu masyarakat untuk mengenali jenis jejjaitan berdasarkan image jejjaitan yang diinputkan ke dalam sistem.

Pada penelitian ini dataset berjumlah 100 buah data, dari 100 buah dataset digunakan sebagai data *training* sebanyak 50 buah data, dan sebagai data *testing* sebanyak 50 buah data. Dari 50 buah data *testing* masing-masing kelas terdapat 5 buah data, sehingga diharapkan dengan pembagian merata pada setiap kelas data *testing* maka pengujian dapat memperoleh hasil objektif terhadap pengujian data. Pembagian data *training* dan data uji dilakukan secara otomatis dengan memilih id data dalam *database* dan dibagi menggunakan fungsi modulus. Kelas data yang digunakan pada penelitian ini sengaja memiliki kemiripan antar satu data dengan lainnya sehingga diharapkan diperoleh ukuran kemampuan sistem klasifikasi dalam membentuk model klasifikasi dengan kelas yang memiliki ciri hampir mirip. Berikut ini adalah tabel hasil pengujian data yang telah memiliki label:

Tabel 11 : Hasil Pengujian Data

Kelas	0	1	2	3	4	5	6	7	8	9
0	5	-	-	-	-	-	-	-	-	-
1	-	4	1	-	-	-	-	-	-	-
2	-	-	5	-	-	-	-	-	-	-
3	2	2	-	0	-	-	-	-	1	-
4	-	1	-	-	4	-	-	-	-	-
5	-	2	-	-	-	3	-	-	-	-
6	-	-	-	-	-	-	4	-	-	1
7	-	1	-	-	-	-	-	4	-	-
8	-	-	-	-	-	-	-	-	5	-
9	-	-	-	-	-	-	-	-	-	5

Keterangan:

Data *Training* : 50 buah.

Data *Testing* : 50 buah.

Data *Testing* terklasifikasi benar = 39 buah.

Data *Testing* terklasifikasi salah = 11 buah.

$$\text{Akurasi} = \frac{b}{n} * 100\% = \frac{39}{50} * 100\% = 78\%$$

Jadi untuk data yang telah memiliki label memiliki tingkat akurasi sebesar 78 %, hal ini menandakan sistem klasifikasi telah berjalan dengan cukup baik.

### 3. Kesimpulan

Berdasarkan semua tahapan dan penelitian Sistem Pengenalan Jenis *Jejjaitan* Untuk Keperluan *Upakara Jejjaitan* Menggunakan Pengolahan Citra Digital dan *Metode Support Vector Machine (SVM)*, maka dapat disimpulkan:

1. Pengolahan citra digital diperlukan untuk mengolah data input sehingga dapat dilakukan ekstraksi fitur yang ada dalam data tersebut. Fitur yang diambil dari data *training* dan data *testing* adalah 7 fitur *Hu-moment invariant* dan nilai bit dari setiap piksel yang telah diolah.
2. Sistem klasifikasi dengan metode Support Vector Machine telah bekerja sesuai dengan yang diharapkan.
3. Akurasi yang klasifikasi dan pengenalan objek cukup baik yaitu sekitar 78 % dengan jumlah data *training* sebanyak 50 buah dan data uji sebanyak 50 buah, dan sekitar 80 % untuk pengujian data baru yang berjumlah 20 data terdiri dari 2 data untuk setiap kelas.
4. *Jejjaitan* adalah objek yang tidak baku (memiliki motif yang sangat beragam pada setiap kelas) data diperlukan *training* yang sangat banyak untuk meningkatkan akurasi klasifikasi sistem.
5. Ukuran image *height weight* yang tidak seimbang, dan letak objek pada *image* juga mempengaruhi hasil klasifikasi

### Daftar Pustaka

- [1]. Krishna, Indra, Sangeeta, December 2010, *SVM-BDT PNN and Fourier Moment Technique for Classification of Leaf Shape*, International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol 3 No 4.
- [2]. Kadir, Nugroho and Santosa, July to Aug 2011, *Leaf Classification Using Shape, Color, and Texture Features*, International Journal of Trends and Technology.
- [3]. Munir, Rinaldi. "Pengolahan Citra Digital", Bandung: Informatika; 2013.
- [4]. C. Gonzales Rafael, and E. Wood Richard, 2008, *Digital Image Processing Third Edition*, New Jersey, Pearson Education.
- [5]. Huang, Leng, *Analysis of HU's Moment Invariants on Image Scaling and Rotation*, IEEE, 2010;
- [6]. Jiawei Han, and Micheline Kamber. *Data Mining: Concepts and Techniques second edition*, San Fransisco, Morgan Kaufmann Publisher; 2006.
- [7]. Sembiring Krisantus. "Penerapan Teknik Support Vector Machine untuk Pendeteksian Intrusi pada Jaringan." Bandung :Institut Teknologi Bandung; 2007.
- [8]. Pressman SR. *Software Engineering A Practitioner's Approach. 5th ed.* New York: Mc Graw Hill; 2001.

- [9]. Garcia-Molina H, Ullman JD, Widom J. *Database System The Complete Book. 2nd ed.* United States of America: Pearson Prentice Hall; 2009.

#### **Biodata Penulis**

**Ni Ketut Dewi Ari Jayanti**, memperoleh gelar Sarjana Teknik (S.T.), Jurusan Teknik Informatika Universitas Gunadarma Jakarta, lulus tahun 2001. Memperoleh gelar Magister Komputer (M.Kom) Program Pasca Sarjana Magister Ilmu Komputer Universitas Indonesia Jakarta, lulus tahun 2008. Saat ini menjadi Dosen di STMIK STIKOM Bali.

**Kadek Dwi Pradnyani Novianti**, memperoleh gelar Sarjana Pendidikan (S.Pd.), Jurusan Pendidikan Teknik Informatika Universitas Pendidikan Ganesha Singaraja, lulus tahun 2012. Memperoleh gelar Magister Engineering (M.Eng) Program Pasca Sarjana Magister Teknik Elektro Universitas Gajah Mada Yogyakarta, lulus tahun 2015. Saat ini menjadi Dosen di STMIK STIKOM Bali.

**I Wayan Sumalya**, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Sistem Informasi STMIK STIKOM Bali, lulus tahun 2016.