

SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN BIBIT UNGGUL SAPI BALI MENGUNAKAN ALGORITMA *SIMPLIFIED SEQUENTIAL MINIMAL OPTIMIZATION* (SSMO) PADA *SUPPORT VECTOR MACHINE* (SVM)

Eugenius Yosep Korsan^{N1}, Arief Andy Soebroto², Imam Cholissodin³

^{1,2,3}Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya

E-mail: eugeniusykn@gmail.com¹, ariefas@ub.ac.id², imamcs@ub.ac.id³

ABSTRACT

Balai Pembibitan Ternak Unggul (BPTU) Sapi Bali di Jembrana, Bali merupakan sebuah tempat pembudidayaan Sapi Bali yang memiliki kualitas unggul. Sapi Bali merupakan jenis sapi yang memiliki ciri khas yang unik. Ciri khas tersebut terletak pada warna kulit Sapi Bali yang mengalami perubahan sesuai dengan jenis kelamin dan usianya. Pemilihan bibit unggul Sapi Bali di BPTU dilakukan dengan berbagai macam cara. Salah satunya melihat pola warna kulit secara langsung yang terdapat pada tubuh Sapi Bali. Proses pemilihan bibit unggul Sapi Bali rentan terjadinya kesalahan yang dilakukan oleh para peternak (*human error*) dikarenakan jumlah Sapi Bali yang banyak di BPTU Sapi Bali. Pemilihan bibit unggul diklasifikasikan ke dalam tiga kelas yaitu Baik (Bibit Unggul), Sedang, Buruk. Untuk itu, perlu dibutuhkan suatu sistem yang mampu menghasilkan klasifikasi bibit unggul Sapi Bali berdasarkan warna kulit yang diambil menggunakan citra digital. Pada sistem tersebut, akan menerapkan algoritma *Simplified Sequential Minimal Optimization* (SSMO) dengan kernel *Radial Basis Function* (RBF) untuk proses *training* data dan metode *One-Against-All* untuk proses klasifikasi berdasarkan fitur rata-rata dari nilai *red*, *green* dan *blue* (RGB). Hasil dari skenario pengujian didapatkan rata-rata tingkat akurasi untuk empat skenario pengujian Sapi Bali Jantan dan Betina sebesar 97.50% dan 67.50%.

Keywords: Klasifikasi, *One-Against-All*, Sapi Bali, *Simplified Sequential Minimal Optimization*, *Support Vector Machine*

1. INTRODUCTION

Sapi Bali (*Bos sondaicus*) merupakan sapi asli dan murni dari Indonesia yang dihasilkan dari domestikasi banteng liar. Penilaian dan pemilihan bibit unggul Sapi Bali

adalah dengan melihat penampilannya (*performance*). Sapi yang terdapat pada Balai Pembibitan Ternak Unggul (BPTU) merupakan jenis sapi yang memiliki ciri khas yang unik. Ciri khas tersebut salah satunya terletak pada warna kulit Sapi Bali yang mengalami perubahan warna sesuai dengan jenis kelamin dan usianya (Soekardono dkk., 2009). Warna kulit Sapi Bali merupakan salah satu faktor penting dalam proses klasifikasi kualitas dan pemilihan bibit unggul Sapi Bali.

Penelitian ini dilakukan untuk mengurangi terjadinya kesalahan yang dilakukan oleh peternak dalam proses klasifikasi kualitas dan pemilihan bibit unggul Sapi Bali. Kesalahan dalam pemilihan bibit unggul Sapi Bali dikarenakan peternak melihat secara langsung pola warna kulit dari Sapi Bali mengingat jumlah sapi yang banyak di BPTU. Pemilihan bibit unggul Sapi Bali akan diklasifikasikan ke dalam tiga kelas antara lain Baik (Bibit Unggul), Sedang, dan Buruk (Penyimpangan) (Anonim, 2012). Diperlukan sebuah sistem yang dapat membantu peternak dalam mengambil keputusan untuk pemilihan bibit unggul Sapi Bali. Sistem tersebut mampu mengklasifikasikan Sapi Bali berdasarkan warna kulit sesuai dengan kelas yang telah ditentukan berbasis citra digital. Fitur yang digunakan dalam proses pengklasifikasian merupakan komponen warna dari warna kulit Sapi Bali yaitu *red*, *green* dan *blue* (RGB) (Munawaroh dan Sutanto, 2010).

Penelitian yang berkaitan dengan penggunaan citra digital sebagai objek utama dan fitur RGB dilakukan oleh Eliyani, Tulus dan Fahmi (2013) yang berjudul “*Pengenalan Tingkat Kematangan Pepaya Paya Rabo Menggunakan Pengolahan Citra Digital Berdasarkan Warna RGB dengan K-Means Clustering*”. Penelitian ini menggunakan metode *K-Mean Clustering* yang digunakan untuk mengelompokkan citra buah pepaya paya rabo berdasarkan tingkat kematangannya

menggunakan fitur *red*, *green*, dan *blue* (RGB). Tingkat kematangan yang dikelompokkan antara lain masak muda, masak mengkal dan masak penuh. Penelitian selanjutnya adalah penelitian yang dilakukan oleh Vadicherla dan Sonawane (2013) yang berjudul “*Decision Support System For Heart Disease Based On Sequential Minimal Optimization In Support Vector Machine*”. Penelitian ini bertujuan untuk memberi keputusan dalam mendiagnosa pasien yang memiliki penyakit jantung dengan mengklasifikasikan menjadi 2 kelas berdasarkan 14 fitur.

Penelitian ini menerapkan algoritma *Simplified Sequential Minimal Optimization* (SSMO) dengan kernel *Radial Basis Function* (RBF) untuk mengklasifikasikan citra Sapi Bali berdasarkan fitur rata-rata dari nilai *red*, *green* dan *blue* (RGB) ke dalam tiga kelas yaitu Baik, Sedang dan Buruk. Penyelesaian klasifikasi *multi-class* dalam penelitian ini akan menggunakan metode *one-against-all*.

2. METODOLOGI

Penelitian ini dilakukan dengan melakukan pengembangan sistem pendukung keputusan pemilihan bibit unggul Sapi Bali menggunakan algoritma *Simplified Sequential Minimal Optimization* (SSMO) yang merupakan pengembangan dari algoritma *Sequential Minimal Optimization* (SMO) (Platt, 1998). Tahapan-tahapan yang dilakukan dalam penelitian ini adalah sebagai berikut.

2.1 Studi Literatur

Studi literatur dilakukan dengan tujuan untuk mempelajari dasar teori dan sumber acuan tentang Sapi Bali, citra digital, algoritma *Simplified SMO*, SMO dan SVM.

2.2 Pengumpulan Data

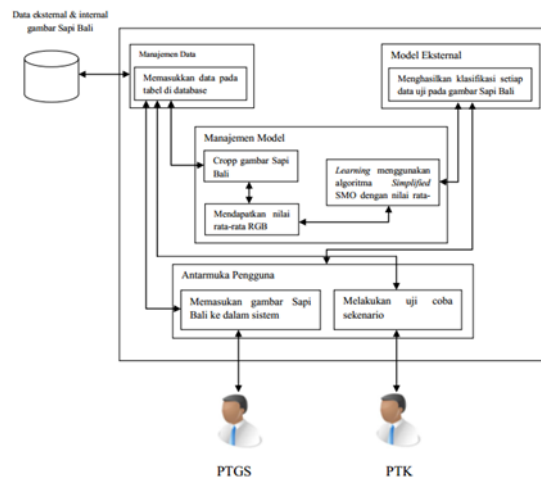
Lokasi penelitian ini di Balai Pembibitan Ternak Unggul yang terletak di kabupaten Jembrana, Bali. Penelitian ini menggunakan data Sapi Bali yang digunakan sebagai acuan untuk pengembangan aplikasi ini. Data yang digunakan 90 data untuk jenis kelamin jantan dan betina pada Sapi Bali.

2.3 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan dari sistem yang akan dibangun.

2.4 Perancangan

Perancangan sistem dibuat berdasarkan hasil pengambilan data dan analisis kebutuhan yang telah dilakukan. Perancangan sistem meliputi perancangan arsitektur SPK pemilihan Spai Bali, perancangan algoritma *Simplified SMO* dan perancangan metode *one-against-all*. Perancangan arsitektur SPK pemilihan bibit unggul Sapi Bali menggunakan algoritma SSMO pada SVM dapat dilihat pada Gambar 1.





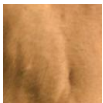
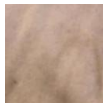


Gambar 1. Arsitektur SPK Pemilihan Bibit Unggul Sapi Bali

Tahap umum SPK pemilihan bibit unggul Sapi Bali pada Gambar 1 yaitu melakukan proses memasukkan *dataset* berupa Gambar Sapi Bali yang dilakukan oleh petugas BPTU. Data tersebut terlebih dahulu akan di *cropp* sebanyak tiga (3) buah dengan ukuran tertentu pada warna kulit bagian samping dari tubuh Sapi Bali sesuai dengan kebutuhan. Tahap selanjutnya peternak akan melakukan skenario uji coba data dengan menginputkan nilai pada parameter yang dibutuhkan dalam proses pelatihan data. Pada tahap pelatihan ini menggunakan fitur rata-rata *red*, *green* dan *blue* (RGB) yang didapatkan dari *cropping* Gambar Sapi Bali dan akan menghasilkan nilai optimal dari α dan *bias* pada tiap data latihserta fungsi keputusan untuk proses klasifikasi. Tahap terakhir dilakukan proses klasifikasi menggunakan metode *one-against-all* untuk menangani klasifikasi *multi-class*.

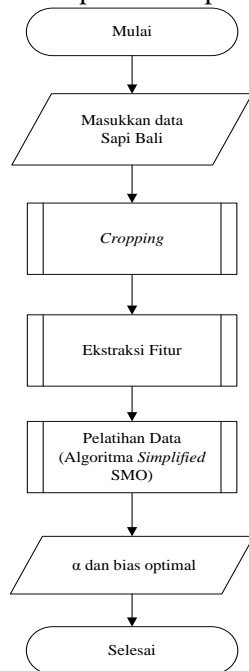
Kriteria warna kulit dalam proses klasifikasi Sapi Bali ditentukan berdasarkan nilai RGB yang terdapat pada Gambar Sapi Bali. Tabel 1 merupakan kriteria warna kulit Sapi Bali yang telah melalui proses *cropping*.

Tabel 1. Kriteria Warna Kulit Sapi Bali

Jenis Kelamin	Klasifikasi		
	Baik	Sedang	Buruk
Jantan			
	R:72;G:65;B:75	R:105;G:73;B:49	R:185;G:167;B:144
Betina			
	R:144;G:110;B:93	R:190;G:134;B:80	R:182;G:154;B:135

Sumber: Observasi

Perancangan diagram algoritma *Simplified SMO* dapat dilihat pada Gambar 2.

Gambar 2. Diagram Alir Algoritma *Simplified SMO*

Penggunaan algoritma *Simplified SMO* dapat digunakan setelah proses ekstraksi fitur untuk mendapatkan fitur *red*, *green* dan *blue* (RGB). Algoritma *Simplified SMO* digunakan dalam proses pelatihan data latih. Pada penelitian ini melakukan dua kali proses pelatihan. Pelatihan pertama dilakukan untuk

kelas Baik vs Buruk sedangkan pada pelatihan ke 2 untuk kelas Baik vs Sedang. Berikut beberapa tahapan yang digunakan dalam algoritma *Simplified SMO* pada pelatihan pertama antara lain (Andrew, 2009)]:

1. Mendapatkan nilai fitur dari rata-rata *red*, *green* dan *blue* (RGB) untuk data uji dan data latih. Berikut adalah contoh data latih dan data uji yang ditunjukkan pada Tabel 2 dan Tabel 3.

Tabel 2. Data Latih

ID Sapi	Fitur			Kelas	Target (y)
	R	G	B		
X1	70	67	55	1	Baik
X2	57	61	68	1	Baik
X3	133	88	50	2	Sedang
X4	164	124	96	2	Sedang
X5	170	130	86	3	Buruk
X6	173	156	125	3	Buruk

Tabel 3. Data Uji

ID Sapi	Fitur			Kelas	Target (y)
	R	G	B		
X7	70	67	55	?	?

2. Inisialisasi awal
Inisialisasi awal $\alpha = 0$, $bias = 0$ dan maksimum iterasi = 1, $C = 1$, $\gamma = 0.001$, $tol1 = 0.01$, dan $tol2 = 0.0001$.
3. Menghitung nilai *dot product* dari masing-masing data latih menggunakan kernel RBF yang ditunjukkan oleh persamaan (1).

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad \gamma > 0(1)$$

Dimana:

$K(x_i, x_j)$ = kernel terhadap dua titik (data).

γ = gamma

$\|x_i - x_j\|^2$ = euclidean distance terhadap dua titik (data)

Berikut contoh perhitungan untuk nilai kernel pada data X1 dan X2.

$$\begin{aligned}
 K(x_1, x_2) &= \exp(-\gamma \|x_1 - x_2\|^2) \\
 &= \exp(-0,001 * ((70 - 57)^2 \\
 &\quad + (67 - 61)^2 + (55 - 88)^2)) \\
 &= \exp(-0,374) \\
 &= 0,69
 \end{aligned}$$

Berdasarkan contoh perhitungan *kernel* diatas maka didapatkan nilai *dot product* untuk data latih ditunjukkan pada Tabel 3.

Tabel 3. Hasil Perhitungan dengan Kernel RBF

K(xi,xj)	X1	X2	X3	X4	X5	X6
X1	1	0.687976912	0.011855066	1.05076E-06	3.28083E-07	6.67545E-11
X2	0.687976912	1	0.00108194	9.19519E-08	1.76241E-08	6.69273E-12
X3	0.011855066	0.00108194	1	0.012613344	0.01192641	7.1457E-06
X4	1.05076E-06	9.19519E-08	0.012613344	1	0.841979173	0.142844308
X5	3.28083E-07	1.76241E-08	0.01192641	0.841979173	1	0.11014033
X6	6.67545E-11	6.69273E-12	7.1457E-06	0.142844308	0.11014033	1
X7	0.003848776	0.00083007	0.228778727	0.155984287	0.082578989	0.001606014

4. Melakukan perhitungan nilai *error* pada data ke-*i* untuk semua data latih dengan menggunakan persamaan (2) dan (3).

$$f(x) = \sum_{i=1}^n \alpha_i y_i K(x, x_i) + b(2) E_i = f(x) - y_i$$

(3)

Dimana:

n = Jumlah data

α_i = *alpha* pada data ke-*i*

y_i = target pada data ke-*i*

b = bias

E_i = *error* pada data ke-*i*

Berikut contoh perhitungan untuk mencari nilai *error* pada perulangan ke-1 dengan data ke-1.

$$\begin{aligned} f(x) &= ((0 * 1 * 1) + (0 * 1 * 0,687976912) \\ &\quad + (0 * 1 * 0,011855066) \\ &\quad + (0 * 1 * 1,05076E - 06) \\ &\quad + (0 * -1 * 3,28083E - 07) \\ &\quad + (0 * -1 * 6,67545E - 11) \\ &\quad + 0 = 0 \end{aligned}$$

$$E_1 = 0 - 1 = -1$$

Berdasarkan perhitungan *error(i)* di atas maka didapatkan nilai *error* pada tiap data training untuk masing-masing perulangan seperti yang ditunjukkan pada Tabel 4.

Tabel 4. Hasil *error* pada data ke-*i*

Perulangan ke-	data ke-i	f(x)	yi	Ei
1	1	0	1	-1
2	2	0	1	-1
3	3	0,001074794	1	-0,99892521
4	4	-0,142844216	1	-1,14284422
5	5	-0,983648048	-1	0,016351952
6	6	-0,983648066	-1	0,016351934
7	1	0,814468848	1	-0,18553115
8	2	1,126492338	1	0,126492338
9	3	0,128253991	1	-0,87174601
10	4	0,141668875	1	-0,85833112
11	5	-0,141668875	-1	0,858331125
12	6	-0,840803758	-1	0,159196242

5. Melakukan pengecekan terhadap kondisi KKT dimana $\alpha_i < C$ dan $\alpha_i > 0$. Hasil dari kondisi KKT berupa nilai *true* atau

false. Hasil kondisi KKT pada tiap data latih untuk masing-masing perulangan dapat dilihat pada Tabel 5.

Tabel 5. Kondisi KKT

Perulangan ke-	Hasil	Perulangan ke-	Hasil
1	True	7	True
2	True	8	True
3	True	9	True
4	True	10	False
5	False	11	False
6	False	12	False

6. Memilih data ke-*j* sebagai data pembandingan secara acak (*random*) yang akan dioptimasi secara bersama-sama dengan data ke-*i*.
7. Menghitung nilai *error* pada data ke-*j* dengan menggunakan persamaan (2) dan (4).

$$E_j = f(x) - y_j \quad (4)$$

Dimana:

y_j = target pada data ke-*j*

E_j = *error* pada data ke-*j*

Berikut contoh perhitungan untuk mencari nilai *error(j)* pada perulangan ke-1 dengan data ke-3.

$$\begin{aligned} f(x) &= ((0 * 1 * 0,011855066) \\ &\quad + (0 * 1 * 0,00108194) \\ &\quad + (0 * 1 * 1) \\ &\quad + (0 * 1 * 0,012613344) \\ &\quad + (0 * -1 * 0,01192641) \\ &\quad + (0 * -1 * 7,1457E - 06) + 0 = 0 \\ E_3 &= 0 - 1 = -1 \end{aligned}$$

Berdasarkan perhitungan *error(j)* di atas maka didapatkan nilai *error* pada tiap data training untuk masing-masing perulangan seperti yang ditunjukkan pada Tabel 6.

Tabel 6. Hasil *error* pada data ke-j

Perulangan ke-	Data ke-j	f(x)	yj	Ej
1	3	0	1	-1
2	6	0	-1	1
3	6	-1	-1	6.69E-12
4	5	-0,11014	-1	0.88986
5	null	null	null	null
6	null	null	null	null
7	5	-0,98365	-1	0.016352
8	4	0,141669	1	-0.85833
9	5	-0,14167	-1	0.858331
10	null	null	null	null
11	null	null	null	null
12	null	null	null	null

8. Menyimpan nilai α_i dan α_j awal ke dalam variabel α_{i_lama} dan α_{j_lama} . Tabel 7 menunjukkan hasil menyimpan nilai alpha ke dalam variabel alpha_lama.

Tabel 7. Nilai alpha lama data i dan j

Perulangan ke-	i	j	Alpha_lam a-i	Alpha_lam a-j
1	1	3	0	0
2	2	6	0	0
3	3	6	0	1
4	4	5	0	0
5	1	null	null	null
6	1	null	null	null
7	1	5	0	1
8	2	4	1	1
9	3	5	0	1
10	1	null	null	null
11	1	null	null	null
12	1	null	null	null

9. Menghitung nilai L dan H menggunakan persamaan (5) dan (6) yang sangat bergantung pada target (y) yang dimiliki pada tiap data latih.

$$L = \begin{cases} \max(0, \alpha_2^{lama} - \alpha_1^{lama}); & \text{Jika } y_i \neq y_j \\ \max(0, \alpha_2^{lama} + \alpha_1^{lama} - C); & \text{Jika } y_i = y_j \end{cases} \quad (5)$$

$$H = \begin{cases} \min(C, C + \alpha_2^{lama} - \alpha_1^{lama}); & \text{Jika } y_i \neq y_j \\ \min(C, \alpha_2^{lama} + \alpha_1^{lama}); & \text{Jika } y_i = y_j \end{cases} \quad (6)$$

Dimana:

L = nilai maksimum

H = nilai minimum

C = batasan *error* bernilai positif

Contoh perhitungan mencari nilai L dan H pada perulangan ke-1 dengan $y(i) = +1$ dan $y(j) = -1$.

$$L = \max(0, \alpha[i] + \alpha[j] - C)$$

$$L = \max(0, 0 - 0 - 1)$$

$$L = 0$$

$$H = \min(C, \alpha[j] + \alpha[i])$$

$$H = \min(1, 0 + 0)$$

$$H = 0$$

Berdasarkan perhitungan di atas maka didapatkan nilai L dan H pada tiap data latih untuk masing-masing perulangan ditunjukkan pada Tabel 8 dan Tabel 9.

Tabel 8. Nilai L dan H (target sama)

Perulangan ke-	L	H	target(y) i	target(y) j
1	0	0	1	1
2	0	0	1	-1
3	0	1	1	-1
4	0	0	1	-1
5	null	null	null	null
6	null	null	null	null
7	0	1	1	-1
8	1	1	1	1
9	0	1	1	1
10	null	null	null	null
11	null	null	null	null
12	null	null	null	null

Tabel 9. Nilai L dan H (target tidak sama)

Perulangan ke-	L	H	target(y) i	target(y) j
1	0	1	1	1
2	0	1	1	-1
3	1	1	1	-1
4	0	1	1	-1
5	null	null	null	null
6	null	null	null	null
7	1	1	1	-1
8	0	1	1	-1
9	1	1	1	-1
10	null	null	null	null
11	null	null	null	null
12	null	null	null	null

10. Menghitung nilai eta (η) dengan persamaan (7).

$$\eta = 2K(x_i, x_j) - K(x_i, x_i) - K(x_j, x_j) \quad (7)$$

Dimana:

$K(x_i, x_i)$ = nilai *kernel* untuk dua titik data

Contoh perhitungan *eta*(η) pada perulangan ke-4 untuk $i = 4$ dan $j = 5$. Nilai $K(x_i, x_j)$, $K(x_i, x_i)$, $K(x_j, x_j)$ didapatkan pada perhitungan menggunakan kernel RBF pada Tabel 3.

$$\eta = 2K(x_i, x_j) - K(x_i, x_i) - K(x_j, x_j)$$

$$= 2 * 0,841979173 - 1,00 - 1,00$$

$$= -3,16041653663000E - 01$$

Berdasarkan contoh perhitungan di atas maka hasil perhitungan nilai *eta* pada masing-masing proses ditunjukkan pada Tabel 10.

Tabel 10. Hasil perhitungan η

Perulangan ke-	Eta	i	j
1	null	1	3
2	-1,99999999998661E+00	2	6
3	null	3	6
4	-3,16041653663000E-01	4	5
5	null	null	null
6	null	null	null
7	null	null	null
8	null	null	null
9	null	null	null
10	null	null	null
11	null	null	null
12	null	null	null

11. Menghitung nilai α_j baru menggunakan persamaan (8) dan (9).

$$\alpha_j = \alpha_j - \frac{y_j(E_i - E_j)}{\eta} \quad (8)$$

Dimana:

y_j = target data ke-j

E_i = error data ke-i

E_j = error data ke-j

Contoh perhitungan *alpha baru* untuk data ke-j pada perulangan ke-4 untuk i = 4 dan j = 5. Nilai E_i dan E_j didapatkan pada Tabel 4 dan Tabel 6.

$$\begin{aligned} \alpha_j &= \alpha_j - \frac{y_j(E_i - E_j)}{\eta} \\ &= 0 - \frac{-1(-1,142844216 + 0,889859688)}{-3,16041653663000E - 01} \\ &= 6,43175948426762 \end{aligned}$$

Berdasarkan contoh perhitungan diatas maka hasil perhitungan alpha baru untuk data ke-j pada masing-masing proses ditunjukkan pada Tabel 11.

Tabel 11. Nilai Alpha Baru data ke-j

Perulangan ke-	alpha_baru ke-j	i	j
1	null	1	3
2	1,0000000000066900E+00	2	6
3	null	3	6
4	6,4317594842676200E+00	4	5
5	null	null	null
6	null	null	null
7	null	null	null
8	null	null	null
9	null	null	null
10	null	null	null
11	null	null	null
12	null	null	null

12. Menentukan batasan maksimum nilai *alpha* data ke-j dengan memotong segmen garis selain batas maksimum dengan menggunakan persamaan (9)

$$\alpha_j = \begin{cases} H, & \text{if } \alpha_j > H; \\ \alpha_j, & \text{if } L < \alpha_j < H; \\ L, & \text{if } \alpha_j < L; \end{cases}$$

(9)

Dimana:

y_j = target data ke-j

α_j = *alpha* data ke-j

Tabel 12 menunjukan hasil maksimum nilai *alpha* pada masing-masing perulangan.

Tabel 12. Nilai Alpha data j pada segmen garis

Perulangan ke-	alpha_baru ke-j	j
1	null	3
2	1	6
3	null	6
4	1	5
5	null	null
6	null	null
7	null	null
8	null	null
9	null	null
10	null	null
11	null	null
12	null	null

13. Mengecek nilai absolut dari $\alpha_{baru}[j] - \alpha_{lama}[j]$ lebih kecil dari nol terpenuhi atau tidak. Hasil dari pengecekan nilai absolut berupa memenuhi atau tidak memenuhi. Tabel 13 menunjukan hasil kondisi pengecekan nilai absolut.

Tabel 13. Kondisi Nilai Absolut

Perulangan ke-	hasil	kondisi
1	null	null
2	1	Tidak Memenuhi
3	null	null
4	1	Tidak Memenuhi
5	null	null
6	null	null
7	null	null
8	null	null
9	null	null
10	null	null
11	null	null
12	null	null

14. Menghitung nilai α_i menggunakan persamaan (10).

$$\alpha_i = \alpha_i + y_i * y_j(\alpha_j^{lama} - \alpha_j) \quad (10)$$

Dimana:

y_i = target pada data ke-i

α_j^{lama} = nilai dari α_j sebelum dilakukan optimasi pada persamaan (8) dan (9).

Contoh perhitungan nilai α_i pada perulangan ke-4 untuk $i = 4$ dan $j = 5$. Nilai α_j didapatkan pada Tabel 13.

$$\begin{aligned}\alpha_i &= \alpha_i + y_i * y_j(\alpha_j^{lama} - \alpha_j) \\ &= 0 + (1 * 1) * (0 - 1) \\ &= 1\end{aligned}$$

Berdasarkan perhitungan di atas didapatkan hasil pada masing-masing perulangan ditunjukkan pada Tabel 14.

Tabel 14 Nilai Alpha Baru Data ke-i

Perulangan ke-	alpha_baru ke-i	i	j
1	null	1	3
2	1	2	6
3	null	3	6
4	1	4	5
5	null	null	null
6	null	null	null
7	null	null	null
8	null	null	null
9	null	null	null
10	null	null	null
11	null	null	null
12	null	null	null

15. Mencari nilai bias 1 dan bias 2 dengan persamaan (11) dan (12).

$$b_1 = b - E_i - y_i(\alpha_i - \alpha_i^{lama})K(x_i, x_i) - y_j(\alpha_j - \alpha_j^{lama})K(x_i, x_j) \quad (11)$$

$$b_2 = b - E_j - y_i(\alpha_i - \alpha_i^{lama})K(\tilde{x}_i, \tilde{x}_j) - y_j(\alpha_j - \alpha_j^{lama})K(x_j, x_j) \quad (12)$$

Dimana:

b = bias

E_i = error pada data ke-i

E_j = error pada data ke-j

$K(x_i, x_j)$ = kernel dua titik data

Contoh perhitungan $bias$ 1 dan 2 untuk data ke-i pada perulangan ke-4 untuk $i = 4$ dan $j = 5$.

$$\begin{aligned}b_1 &= b - E_i - y_i(\alpha_i - \alpha_i^{lama})K(x_i, x_i) - y_j(\alpha_j - \alpha_j^{lama})K(x_i, x_j) \\ &= 3,92827E - 17 - (-1,142844216) - 1 * (1 - 0) * 1 - 1 \\ &\quad * (1 - 0) * 0,841979173 \\ &= 0,984823389\end{aligned}$$

$$\begin{aligned}b_2 &= b - E_i - y_i(\alpha_i - \alpha_i^{lama})K(x_i, x_j) - y_j(\alpha_j - \alpha_j^{lama})K(x_j, x_j) \\ &= 3,92827E - 17 - 0,889859688 - 1 * (1 - 0) * 0,841979173 \\ &\quad - 1 * (1 - 0) * 1 \\ &= -0,731838861\end{aligned}$$

Berdasarkan contoh perhitungan diatas maka didapatkan hasil nilai bias 1 dan bias 2 untuk masing-masing perulangan yang ditunjukkan pada Tabel 15.

Tabel 15. Nilai Bias 1 dan Bias 2

Perulangan ke-	bias_1	bias_2
1	null	null
2	6,69273E-12	-6,69265E-12
3	null	null
4	0,984823389	-0,731838861
5	null	null
6	null	null
7	null	null
8	null	null
9	null	null
10	null	null
11	null	null
12	null	null

16. Mencari nilai bias menggunakan persamaan (13)

$$b = \begin{cases} b_1, & \text{if } 0 < \alpha_i < C \\ b_2, & \text{if } 0 < \alpha_j < C \\ \frac{(b_1 + b_2)}{2}, & \text{lainnya} \end{cases} \quad (13)$$

Berikut contoh perhitungan $bias_baru$ pada perulangan ke-4. Diketahui $\alpha_j = 1$, $\alpha_i = 1$, $C = 1$ sehingga bias baru pada perulangan ke-4 menggunakan kondisi ke-3.

$$\begin{aligned}b &= \frac{b_1 + b_2}{2} \\ &= \frac{0,984823389 + (-0,731838861)}{2} \\ &= 0,126492264\end{aligned}$$

Berdasarkan contoh perhitungan diatas maka didapatkan nilai bias (b) pada setiap perulangan yang ditunjukkan pada Tabel 16.

Tabel 16. Nilai Bias

Perulangan ke-	bias_baru
1	null
2	3,92827E-17
3	null
4	0,126492264
5	null
6	null
7	null
8	null
9	null
10	null
11	null
12	null

Langkah-langkah diatas dilakukan sampai nilai α pada semua data latih tidak memenuhi kondisi KKT dan tidak mengalami perubahan. Tabel 17 menunjukan nilai alpha dan bias yang optimal dari proses pelatihan data latih.

Tabel 17. Nilai Alpha dan Bias Optimal
Pelatihan Pertama

Alpha 1	0
Alpha 2	1
Alpha 3	0
Alpha 4	1
Alpha 5	1
Alpha 6	1
bias	0,126492264

Berdasarkan nilai *alpha* dan *bias* dari pelatihan pertama (Baik vs Buruk) maka didapatkan fungsi keputusan yang sesuai dengan persamaan (14) untuk pelatihan diatas adalah sebagai berikut:

$$f(x_d) = \text{sign}(\sum_{i=1}^n \alpha_i y_i K(x_i, x_d) + b) \quad (14)$$

$$f^1(x_{uji}) = \text{sign}((0 * 1 * K(x_i, x_{uji})) + (1 * 1 * K(x_i, x_{uji})) + (0 * 1 * K(x_i, x_{uji})) + (1 * 1 * K(x_i, x_{uji})) + (1 * -1 * K(x_i, x_{uji})) + (1 * -1 * K(x_i, x_{uji})) + (0,126492264))$$

Tahapan-tahapan pelatihan kedua (Baik vs Sedang) sama seperti pelatihan pertama (Baik vs Buruk) maka didapatkan nilai optimal alpha dan bias pada pelatihan kedua ditunjukkan pada Tabel 18.

Tabel 18. Nilai Alpha dan Bias Optimal
Pelatihan Kedua

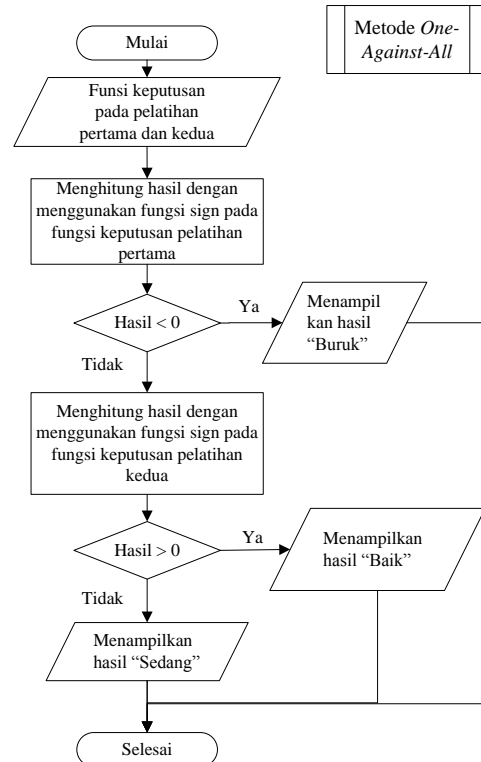
Alpha 1	0,761346425
Alpha 2	0,730016123
Alpha 3	0,753859558
Alpha 4	0,73750299
bias	-0,252989187

Berdasarkan nilai *alpha* dan *bias* dari pelatihan kedua (Baik vs Sedang) maka didapatkan fungsi keputusan yang sesuai dengan persamaan (14) untuk pelatihan diatas adalah sebagai berikut:

$$f(x_d) = \text{sign}(\sum_{i=1}^n \alpha_i y_i K(x_i, x_d) + b)$$

$$f^2(x_{uji}) = \text{sign}((0,761346425 * 1 * K(x_i, x_{uji})) + (0,730016123 * 1 * K(x_i, x_{uji})) + (0,753859558 * -1 * K(x_i, x_{uji})) + (0,73750299 * -1 * K(x_i, x_{uji})) + (-0,252989187))$$

Perancangan diagram alir metode *one-against-all* dapat dilihat pada Gambar 3.



Gambar 3. Diagram Alir Metode *One-Against-All*

Penggunaan metode *one-against-all* untuk melakukan proses klasifikasi ke dalam kelas baik, sedang atau buruk pada data Sapi Bali. Tahapan penggunaan metode *one-against-all* sebagai berikut:

1. Mendapatkan fungsi keputusan yang dihasilkan proses peltihan pertama dan kedua.
2. Memasukan nilai kernel untuk data uji yang didapatkan pada Tabel 3.
3. Melakukan proses sesuai dengan diagram alir pada Gambar 3.

$$f^1(x_{uji}) = \text{sign}((0 * 1 * 0,003848776) + (1 * 1 * 0,00083007) + (0 * 1 * 0,228778727) + (1 * 1 * 0,155984287) + (1 * -1 * 0,082578989) + (1 * -1 * 0,001606014) + (0,126492264))$$

$$f^1(x_{uji}) = \text{sign}(0,1991216) = 1$$

Berdasarkan perhitungan pelatihan pertama didapatkan nilai +1 sehingga dilakukan lagi perhitungan pada fungsi keputusan pada pelatihan kedua.

$$f^2(x_{uji}) = \text{sign}((0,761346425 * 1 * 0,003848776) \\ + (0,730016123 * 1 * 0,00083007) \\ + (0,753859558 * -1 \\ * 0,228778727) \\ + (0,73750299 * -1 \\ * 0,155984287) \\ + (-0,252989187)) \\ f^2(x_{uji}) = \text{sign}(-0,536958878) = -1$$

Pada perhitungan fungsi pelatihan kedua didapatkan hasil **-1** sehingga untuk data uji tersebut dapat diklasifikasikan kedalam kelas **sedang**.

2.5 Implementasi

Implementasi sistem dilakukan dengan mengacu kepada perancangan sistem. Pada tahapan implmentasi menjelaskan proses pembuatan algoritma *Simplified SMO* menggunakan bahasa JAVA dan menggunakan MySQL sebagai databasenya.

2.6 Pengujian

Pengujian yang dilakukan adalah pengujian tingkat akurasi data dengan membandingkan keluaran dari sistem dengan hasil pakar menggunakan algoritma *Simplified SMO*. Skenario pengujian yang dilakukan pada penelitian ini adalah pengujian untuk mengetahui pengaruh ukuran *cropping* dan rasio perbandingan jumlah data latih dan data uji, pengujian untuk mengetahui pengaruh nilai parameter C, pengujian untuk mengetahui pengaruh nilai parameter *gamma* (γ), pengujian untuk mengetahui pengaruh jumlah iterasi.

3 HASIL PEMBAHASAN

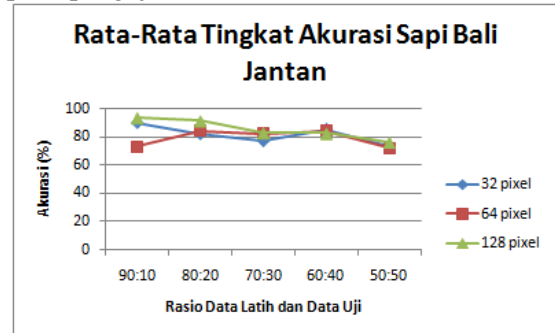
3.1 Pengujian Tingkat Akurasi

Pengujian tingkat akurasi pada penelitian ini dengan membandingkan data yang sebenarnya yang diperoleh di lapangan dengan data yang telah diproses menggunakan algoritma *Simplified SMO*

3.1.1 Hasil Pengujian Tingkat Akurasi Terhadap Pengaruh Ukuran dan Rasio Perbandingan Data Latih dan Data Uji

Pengujian dilakukan untuk mengetahui ukuran *cropping* dan rasio perbandingan data latih dan data uji yang terbaik berdasarkan rata-rata tingkat akurasi yang dihasilkan. Ukuran *cropping* yang akan diuji yaitu 32×32 pixel, 64×64 pixel dan 128×128 pixel serta rasio perbandingan 90:10, 80:20, 70:30, 60:40 dan 50:50 dari masing-masing jenis kelamin dan

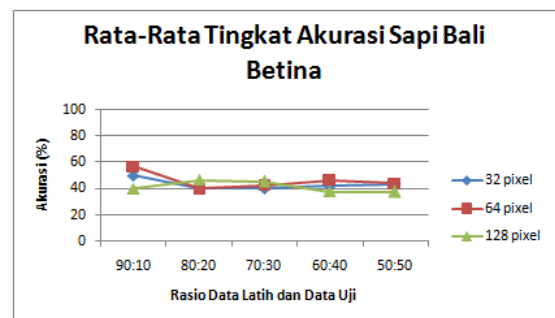
kelas. Pengambilan data latih dan data uji dilakukan secara acak (*random*). Pengujian ini menggunakan beberapa nilai parameter yang tetap yaitu $C = 1$, $\text{gamma} = 0.0001$ dan jumlah iterasi = 1. Tingkat akurasi yang dihasilkan merupakan rata-rata dari lima kali percobaan pada pengujian ini.



Gambar 4. Grafik Rata-Rata Tingkat Akurasi Sapi Bali Jantan Terhadap Ukuran *Cropping* dan Rasio Perbandingan Data Latih dan Data Uji

Berdasarkan grafik pada Gambar 4 diketahui bahwa rata-rata tingkat akurasi tertinggi sebesar 93.33 %. Tingkat akurasi tersebut terdapat pada perbandingan rasio 90:10 dan ukuran *cropping* 128 pixel. Dengan demikian ukuran *cropping* 128 dan rasio perbandingan 90:10 untuk data latih dan data uji dan akan digunakan untuk skenario pengujian kedua pada Sapi Bali Jantan.

Gambar 5 merupakan grafik hasil pengujian rata-rata tingkat akurasi terhadap pengaruh ukuran *cropping* dan rasio perbandingan data latih dan data uji pada Sapi Bali Betina.



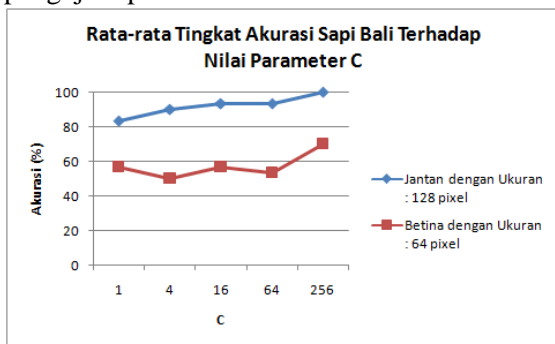
Gambar 5. Grafik Rata-Rata Tingkat Akurasi Sapi Bali Betina Terhadap Ukuran *Cropping* dan Rasio Perbandingan Data Latih dan Data Uji

Berdasarkan grafik pada Gambar 12 diketahui bahwa rata-rata tingkat akurasi tertinggi sebesar 56.67 %. Tingkat akurasi tersebut terdapat pada perbandingan rasio

90:10 dan ukuran *cropping* 64 px. Dengan demikian ukuran *cropping* 64 px dan rasio perbandingan 90:10 untuk data latih dan data uji dan akan digunakan untuk skenario pengujian kedua pada Sapi Bali Betina.

3.1.2 Hasil Pengujian Tingkat Akurasi Terhadap Pengaruh Nilai Parameter C

Pengujian dilakukan untuk mengetahui nilai parameter C yang terbaik berdasarkan rata-rata tingkat akurasi yang dihasilkan. Nilai C yang akan diuji yaitu 1, 4, 16, 64 dan 256. Pengujian ini menggunakan beberapa nilai parameter yang tetap yaitu $\gamma = 0.0001$, jumlah iterasi = 1. Pada Sapi Bali jantan menggunakan ukuran *cropping* 128×128 pixel dan rasio 90:10 sedangkan Sapi Bali betina menggunakan ukuran *cropping* 64×64 pixel dan rasio 90:10 yang didapatkan dari pengujian pertama.



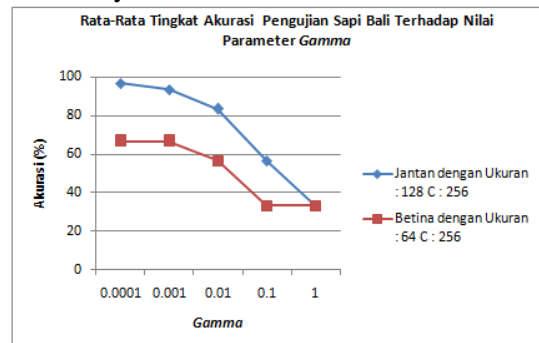
Gambar 6. Grafik Rata-Rata Tingkat Akurasi Sapi Bali Terhadap Nilai Parameter C

Berdasarkan grafik pada Gambar 6 diketahui bahwa rata-rata tingkat akurasi tertinggi sebesar 100 % pada Sapi Bali jantan dengan nilai $C = 256$ dan rata-rata tingkat akurasi tertinggi pada Sapi Bali betina sebesar 70 % dengan nilai $C = 256$. Dengan demikian tingkat akurasi terbaik pada Sapi Bali jantan dan betina akan di gunakan untuk pengujian ketiga.

3.1.3 Hasil Pengujian Tingkat Akurasi Terhadap Pengaruh Nilai Parameter γ

Pengujian dilakukan untuk mengetahui nilai parameter γ yang terbaik berdasarkan tingkat akurasi yang dihasilkan. Nilai γ yang akan diuji yaitu 0.0001, 0.001, 0.01, 0.1 dan 1. Pengujian ini menggunakan nilai parameter yang tetap yaitu jumlah iterasi = 1. Pada Sapi Bali jantan menggunakan ukuran *cropping* 128×128

pixel, rasio 90:10 dan $C = 256$ sedangkan Sapi Bali betina menggunakan ukuran *cropping* 64×64 pixel, rasio 90:10 dan $C = 256$ yang didapatkan dari pengujian sebelumnya.

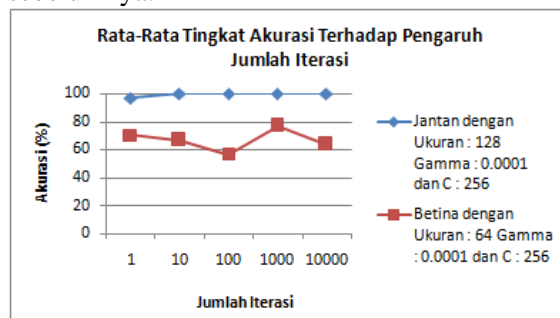


Gambar 7. Grafik Rata-Rata Tingkat Akurasi Sapi Bali Terhadap Nilai Parameter γ

Berdasarkan grafik Gambar 7 diketahui bahwa rata-rata tingkat akurasi tertinggi sebesar 96.67 % pada Sapi Bali jantan dengan nilai $\gamma = 0.0001$. Pada pengujian Sapi Bali betina terdapat dua buah tingkat akurasi terbaik yang sama yaitu pada nilai $\gamma = 0.001$ dan 0.0001 dengan tingkat akurasi sebesar 66.67%. Untuk pengujian selanjutnya pada Sapi Bali betina dan jantan digunakan nilai $\gamma = 0.0001$ pada pengujian ke 4.

3.1.4 Hasil Pengujian Tingkat Akurasi Terhadap Pengaruh Jumlah Iterasi

Pengujian dilakukan untuk mengetahui jumlah iterasi yang terbaik berdasarkan tingkat akurasi yang dihasilkan. Jumlah Iterasi yang akan diuji yaitu 1, 10, 100, 1000 dan 10000. Pada Sapi Bali jantan menggunakan ukuran *cropping* 128×128 pixel, rasio 90:10, $C = 256$ dan $\gamma = 0.0001$ sedangkan Sapi Bali betina menggunakan ukuran *cropping* 64×64 pixel, rasio 90:10, $C = 256$ dan $\gamma = 0.0001$ yang didapatkan dari pengujian sebelumnya.

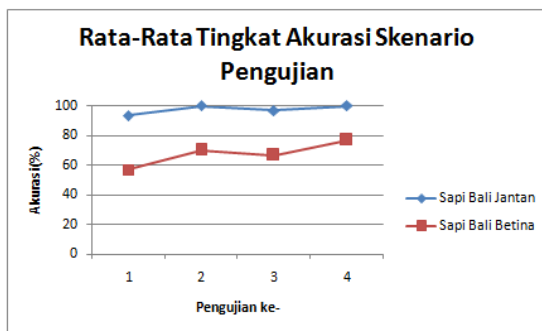


Gambar 8. Grafik Rata-Rata Tingkat Akurasi Sapi Bali Terhadap Jumlah Iterasi

Berdasarkan grafik pada Gambar 8 diketahui bahwa rata-rata tingkat akurasi tertinggi sebesar 100 % pada Sapi Bali jantan dengan jumlah iterasi = 10-1000. Pada pengujian Sapi Bali betina rata-rata tingkat akurasi tertinggi terdapat pada jumlah iterasi = 1000 sebesar 76.67%.

3.2 Analisis Tingkat Akurasi

Pada analisis hasil tingkat akurasi akan dijelaskan mengenai analisa hasil terhadap empat skenario pengujian yang dilakukan menggunakan algoritma *Simplified SMO*. Berdasarkan hasil rata-rata tingkat akurasi keempat skenario pengujian didapatkan rata-rata akurasi sebesar 97.50% dan 67.50% untuk skenario yang dilakukan pada Sapi Bali jantan dan Sapi Bali betina. Grafik rata-rata tingkat akurasi tertinggi tiap skenario untuk Sapi Bali jantan dan Sapi Bali betina dapat dilihat pada Gambar 16.



Gambar 9. Grafik Rata-Rata Tingkat Akurasi Tertinggi Tiap Skenario Pengujian Sapi Bali

Hasil rata-rata tingkat akurasi tertinggi Sapi Bali jantan dan Sapi Bali betina pada pengujian tingkat akurasi terhadap pengaruh ukuran *cropping* dan rasio perbandingan data latih dan data uji sebesar 93.33 % dan 56.67 %. Pada Sapi Bali jantan, ukuran *cropping* dan rasio perbandingan data latih dan data uji yaitu 128 px dan 90:10. Pada Sapi Bali betina, ukuran *cropping* dan rasio perbandingan data latih dan data uji yaitu 64 px dan 90:10. Hal ini dapat disimpulkan bahwa dalam proses klasifikasi yang menggunakan data latih lebih besar dari data uji akan menghasilkan akurasi yang lebih baik. Sedangkan pada ukuran *cropping* terjadi perbedaan antara Sapi Bali jantan dan betina ini disebabkan pada proses pengambilan warna kulit Sapi Bali terdapat *noise* berupa kotoran, hewan (lalat), cahaya yang terlalu terang dan bayangan (*shadow*).

Pada pengujian kedua Sapi Bali jantan dan Sapi Bali betina untuk tingkat akurasi terhadap pengaruh nilai parameter C memiliki rata-rata tingkat akurasi tertinggi sebesar 100% dan 70%. Nilai parameter $C = 256$ merupakan nilai parameter yang terbaik pada Sapi Bali jantan dan Sapi Bali betina. Nilai C digunakan untuk memperkecil *error* pada data latih saat proses pelatihan menggunakan algoritma *Simplified SMO*. Jika nilai C semakin besar maka akan memperkecil *error* data latih pada proses pelatihan sehingga akan menghasilkan tingkat akurasi yang baik pada proses klasifikasi.

Pada pengujian ketiga Sapi Bali jantan dan Sapi Bali betina untuk tingkat akurasi terhadap pengaruh nilai parameter γ memiliki rata-rata tingkat akurasi tertinggi sebesar 96.67% dan 66.67%. Nilai parameter γ terbaik pada Sapi Bali jantan yaitu 0.0001. Nilai parameter γ yang digunakan pada fungsi kernel *Radial Basis Function* (RBF) untuk menentukan kedekatan antar dua titik data latih. Jadi semakin kecil nilai γ maka kedekatan antar dua titik data akan semakin besar sehingga memudahkan untuk menemukan garis pemisah (*hyperplane*) antar dua kelas. Hal ini dapat dilihat dari hasil tingkat akurasi pada nilai $\gamma < 0.01$ lebih baik dari pada nilai $\gamma > 0.01$.

Pada pengujian keempat Sapi Bali jantan dan Sapi Bali betina untuk tingkat akurasi terhadap pengaruh jumlah iterasi memiliki rata-rata tingkat akurasi tertinggi sebesar 100% dan 76.67%. Berdasarkan grafik pada Gambar 8 rata-rata tingkat akurasi tertinggi sebesar 100% untuk jumlah iterasi = 10 - 10000 dan rata-rata tingkat akurasi terendah sebesar 96.67% pada jumlah iterasi = 1 pada Sapi Bali jantan sedangkan berdasarkan grafik pada Gambar 8 Sapi Bali betina memiliki rata-rata tingkat akurasi tertinggi sebesar 76.67% pada jumlah iterasi = 1000 dan rata-rata tingkat akurasi terendah sebesar 56.67% pada jumlah iterasi = 100 tetapi pada jumlah iterasi = 10000 rata-rata tingkat akurasi turun menjadi 63.33%. Hal ini dapat disimpulkan bahwa jumlah iterasi pada penggunaan algoritma *Simplified SMO* tidak memiliki pengaruh yang signifikan terhadap proses klasifikasi.

4 PENUTUP

4.1 Kesimpulan

Berdasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan pada sistem pendukung keputusan pemilihan bibit unggul Sapi Bali menggunakan algoritma *Simplified Sequential Minimal Optimization* (SSMO) pada *Support Vector Machine* (SVM) maka dapat diambil kesimpulan bahwa algoritma *Simplified Sequential Minimal Optimization* (SSMO) menggunakan kernel RBF yang digunakan untuk proses pelatihan data dan metode *One-Against-All* untuk proses klasifikasi *multi-class* layak untuk digunakan untuk pemilihan bibit unggul Sapi Bali berdasarkan warna kulit. Pada Sapi Bali jantan penggunaan perbandingan rasio data latih dan data uji 90:10, ukuran *cropping* 128×128 px, nilai parameter $C = 256$, nilai parameter $\Gamma = 0.0001$ dan jumlah iterasi = 10-10000. Sedangkan Sapi Bali betina penggunaan perbandingan rasio data latih dan data uji 90:10, ukuran *cropping* 64×64 px, nilai parameter $C = 256$, nilai parameter $\Gamma = 0.0001$ dan jumlah iterasi = 1000.

4.2 Saran

Saran yang dapat diberikan untuk pengembangan penelitian ini selanjutnya antara lain:

1. Diharapkan untuk pengambilan data Gambar Sapi Bali di lapangan menggunakan alat bantu (tele/kamera) yang berkualitas bagus sehingga dapat memudahkan dan memberikan hasil Gambar Sapi Bali yang lebih baik.
2. Dalam proses pelatihan data dapat menggunakan algoritma selain *Simplified SMO* seperti algoritma *Chunking*, *Osuna*, *Least Square*, dan *Sequential Training*.
3. Menggunakan berbagai macam jenis kernel seperti *polynomial*, *Sigmoid* dan *linear*.

DAFTAR PUSTAKA

- ANONIM.2012."Materi Petunjuk Standard Penilaian Performance Sapi Bali Di BPTU Sapi Bali". Kementerian Pertanian Direktorat Jendral Peternakan dan Kesehatan Hewan. Balai Pembibitan Ternak Unggul Sapi Bali.
- ELIYANI, TULUS, DAN FAHMI. 2013. "Pengenalan Tingkat Kematangan Buah Pepaya Paya Rabo Menggunakan Pengolahan Citra Digital Berdasarkan Warna RGB dengan K-Means Clustering", Seminar Nasional Teknologi Informasi dan Komunikasi (SNASTIKOM), Sumatera Utara.
- MUNAWAROH, S., SUTANTO, F. A. 2010. "Pengolahan Citra Digital untuk Identifikasi Uang Kertas", Jurnal Teknologi Informasi DINAMIK, Volume XV, No.1, hal 34-40.
- NG, ANDREW. 2009. Cs229 *The Simplified SMO Algorithm*. <http://cs229.stanford.edu/materials/smo.pdf> [5 Juni 2014].
- PLATT, JC. 1998. "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machine". Microsoft Research.
- SOEKARDONO., ARMAN., CHAIRUSSYHUR., KASIP, LALU M. 2009. "Identifikasi Grade Sapi Bali Betina Bibit dan Koefisien Reproduksi Sapi Betina di Propinsi Nusa Tenggara Barat", Buletin Peternakan, Vol.33(2), hal 74-80, Mataram.
- VADICHERLA, DEEPTI., SONAWANE, SHEETAL. 2013. "Decision Support System For Heart Disease Based On Sequential Minimal Optimization In Support Vector Machine". International Journal of Engineering Sciences & Emerging Technologies (IJESSET), Volume 2, Issue 2, pp: 19-26, India