

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

**SKRIPSI  
KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI**

Diajukan untuk memenuhi persyaratan  
memperoleh gelar Sarjana Komputer



Disusun Oleh :

**Eugenius Yosep Korsan Nuhayanan**

**NIM. 105090603111005**

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN**

**PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**

**UNIVERSITAS BRAWIJAYA**

**MALANG**

**2014**

LEMBAR PERSETUJUAN

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

**SKRIPSI  
KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI**

Diajukan untuk memenuhi persyaratan  
memperoleh gelar Sarjana Komputer



Disusun Oleh :

**Eugenius Yosep Korsan Nuhayanan**

**NIM. 105090603111005**

**Skripsi ini telah diperiksa dan disetujui oleh  
Dosen Pembimbing**

**Pembimbing I**

**Pembimbing II**

**Arief Andy Soebroto., S.T., M.Kom.**  
**NIP. 19720425 199903 1 002**

**Imam Cholissodin., S.Si., M.Kom.**  
**NIK. 850719 16 1 1 0422**

**LEMBAR PENGESAHAN**

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

**SKRIPSI**

Disusun Oleh :

**Eugenius Yosep Korsan Nuhyanan**

**NIM. 105090603111005**

Skripsi ini telah diuji dan dinyatakan lulus

**4 Desember 2014**

**Dosen Penguji I**

**Dosen Penguji II**

**Suprpto, ST., MT.**  
**NIP. 19710727 199603 1 001**

**Rekyan Regasari Mardi Putri, ST., MT.**  
**NIK. 770414 06 1 2 0253**

**Dosen Penguji III**

**Mochammad Hannats Hanafi I, S.ST., MT.**  
**NIK. 201405 881229 1 1 001**

Mengetahui,  
Ketua Program Studi Teknik Informatika / Ilmu Komputer

**Drs. Marji, MT**  
**NIP. 19670801 199203 1001**

## LEMBAR PERNYATAAN

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata dalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 4 Desember 2014

**Eugenius Yosep Korsan Nuhayanan**  
**105090603111005**

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Tuhan yang Maha Esa, yang telah melimpahkan berkat dan anugerah-Nya kepada penulis sehingga penulis dapat menyelesaikan skripsi ini yang berjudul **“SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN BIBIT UNGGUL SAPI BALI MENGGUNAKAN ALGORITMA SIMPLIFIED SEQUENTIAL MINIMAL OPTIMIZATION (SSMO) PADA SUPPORT VECTOR MACHINE (SVM)”**

Pada penyusunan Skripsi ini tidak semata-mata hasil kerja penulis sendiri, melainkan juga berkat dan bimbingan dan dorongan dari pihak-pihak yang telah membantu, baik secara materi maupun non materi. Maka dari itu penulis ingin mengucapkan banyak terima kasih yang tak terhingga serta penghargaan yang setinggi-tingginya kepada orang-orang yang telah membantu penulis secara langsung maupun tidak langsung kepada yang terhormat:

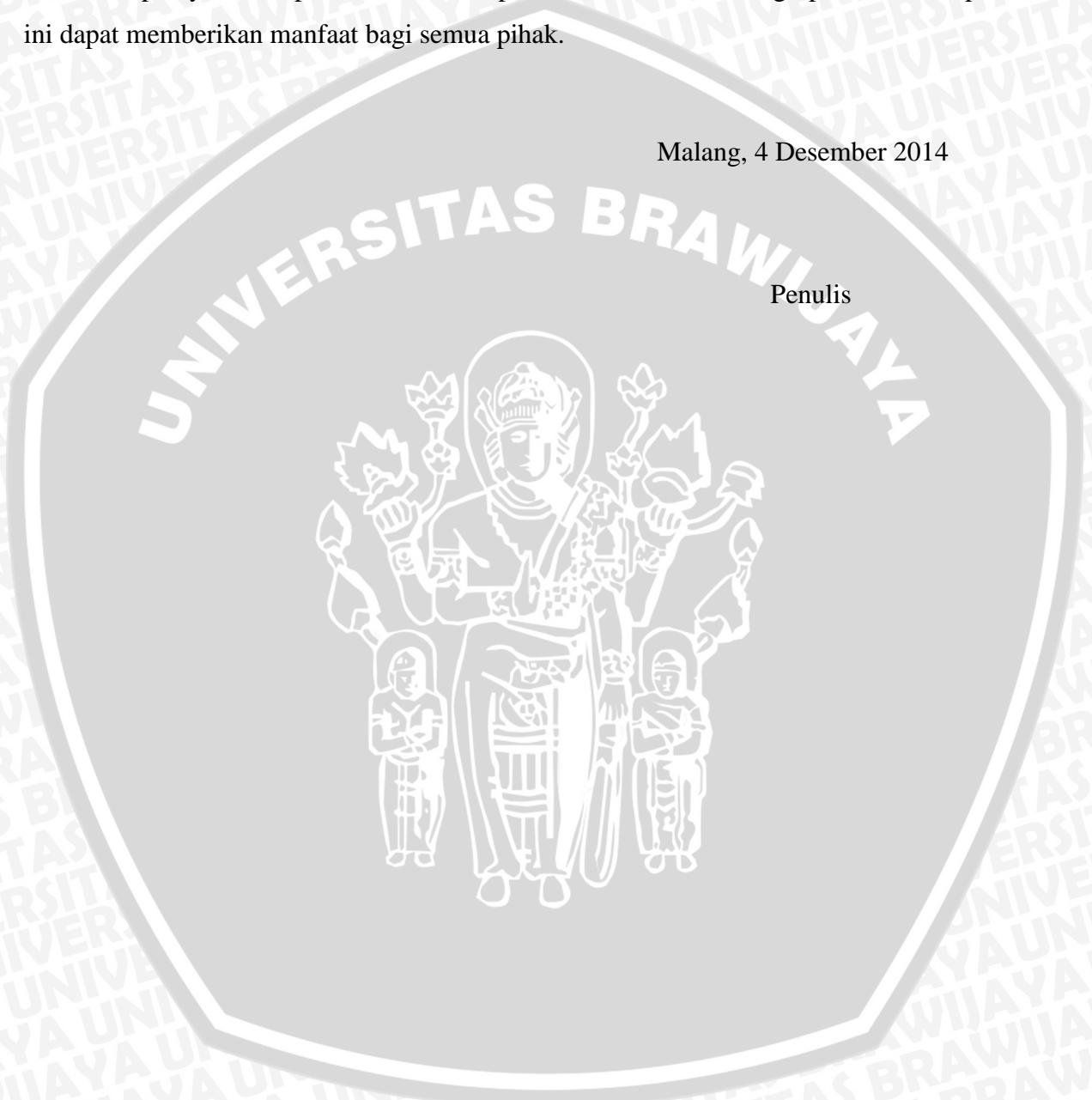
1. Kedua orang tua penulis Carolus Y Nuhayanan dan Yayuk Sulis yang telah memberikan doa, nasehat dan kasih sayang dan kesabarannya dalam membesarkan dan mendidik penulis.
2. Saudaraku FX. Hadisumarta Nuhayanan yang telah membantu dan memberikan semangat dalam pengerjaan skripsi.
3. Drs. Marji, M.T selaku Ketua Prodi Informatika / Ilmu Komputer yang telah memberikan kesempatan kepada penulis untuk menyelesaikan skripsi ini.
4. Arief Andy Soebroto, S.T.,M.Kom dan Imam Cholissodin, S.Si.,M.Kom selaku dosen pembimbing skripsi yang telah membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
5. Segenap bapak dan ibu dosen program studi Informatika / Ilmu Komputer beserta seluruh staff administrasi yang telah membantu selama perkuliahan.
6. Teman – teman Ilmu Komputer angkatan 2010 yang telah memberikan masukan dan inspirasi kepada penulis selama menempuh studi dan menyelesaikan skripsi ini.
7. Dyah ika F. yang selalu memberikan semangat dalam pengerjaan Skripsi.
8. Anggota tim Sapi Bali Guruh P.W.P alias klobot, Danny S alias Mbahh, dan Indro E alias Ndroo

9. Piaraan (Boni dan Bruno) yang senantiasa menemani dan memberikan inspirasi dalam pengerjaan skripsi ini.

Penulis menyadari bahwa penulisan skripsi ini jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan masukan berupa saran dan kritik dari semua pihak demi tercapainya kesempurnaan dalam skripsi ini. Akhir kata semoga penulisan skripsi ini dapat memberikan manfaat bagi semua pihak.

Malang, 4 Desember 2014

Penulis



## ABSTRAK

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). Kombinasi parameter terbaik untuk Sapi Bali Jantan yaitu rasio data latih dan data uji adalah 90:10, ukuran *cropping* 128×128 px, *Complexity* (C) 64, *Gamma* ( $\gamma$ ) 0.001 dan jumlah iterasi 10, sedangkan untuk Sapi Bali Betina yaitu rasio data latih dan data uji adalah 90:10, ukuran *cropping* 128×128 px, *Complexity* (C) 4, *Gamma* ( $\gamma$ ) 0.01 dan jumlah iterasi 1000. Hasil dari skenario pengujian didapatkan tingkat akurasi untuk Sapi Bali Jantan dan Betina sebesar 93.33% dan 80%.

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

## ABSTRACT

*Livestock Breeding Center for Excellence (BPTU) Bali Cattle in Jembrana, Bali is a Bali cattle breeding places which have superior quality. Bali cattle is a type of cow that has a unique characteristic. The distinctive feature lies in the Bali cattle skin color changes according to gender and age. Selection of seeds Bali Cattle in BPTU be done in various ways. One of them saw directly the skin color pattern found on the body of Bali cattle. The process of selecting seeds Bali Cattle vulnerable occurrence of errors made by the breeder (human error) due to the many number of Bali cattle in Bali Cattle BPTU. Selection of seeds classified into three classes, namely Good (Superior Seeds), Average, Poor. For that, we need a system needs to be capable of producing seeds Bali Cattle classification based on skin color were taken using a digital image. In such a system, will apply the Simplified Sequential Minimal Optimization algorithm (SSMO) with kernel Radial Basis Function (RBF) to the training data and methods One-Against-All for the classification based on the features of the average of the value of red, green and blue (RGB ). The combination of the best parameters for Bali cattle Males is ratio of training data and test data are 90:10, the size of cropping 128×128 px, Complexity (C) 64, Gamma ( $\gamma$ ) 0.0001 and the number of iterations are 10, while the females for Bali cattle is ratio of training data and test data are 90:10, the size of cropping 128×128 px, Complexity (C) 4, Gamma ( $\gamma$ ) 0.01 and number of iterations are 1000. The results obtained from testing scenarios for accuracy Bali Cattle Male and Female by 93.33% and 80%.*

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



DAFTAR ISI

<b>LEMBAR PERSETUJUAN</b> .....	<b>i</b>
<b>LEMBAR PENGESAHAN</b> .....	<b>ii</b>
<b>LEMBAR PERNYATAAN</b> .....	<b>iii</b>
<b>KATA PENGANTAR</b> .....	<b>iv</b>
<b>ABSTRAK</b> .....	<b>vi</b>
<b>ABSTRACT</b> .....	<b>vii</b>
<b>DAFTAR ISI</b> .....	<b>viii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xii</b>
<b>DAFTAR TABEL</b> .....	<b>xvi</b>
<b>DAFTAR LAPMPIRAN</b> .....	<b>xix</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	3
1.5 Manfaat .....	4
1.6 Sistematika Penulisan .....	4
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>6</b>
2.1 Kajian Pustaka.....	6
2.2 Sapi Bali .....	12
2.3 Sistem Pendukung Keputusan (SPK).....	17
2.3.1 Karakteristik dan Kemampuan SPK .....	17
2.3.2 Komponen SPK.....	18
2.4 Citra Digital.....	19
2.4.1 Citra Biner .....	20
2.4.2 Citra <i>Grayscale</i> .....	20
2.4.3 Citra Warna .....	21
2.5 Klasifikasi .....	21
2.6 <i>Support Vector Machine (SVM)</i> .....	22
2.6.1 <i>SVM Linear</i> .....	22
2.6.2 <i>SVM NonLinear</i> .....	26



2.7	<i>Multi-Class SVM</i> .....	28
2.7.1	Metode <i>one-against-all</i> .....	28
2.7.2	Metode <i>one-against-one</i> .....	29
2.7.3	Metode <i>Directed Acyclic Graph Support Vector Machine (DAGSVM)</i> .....	31
2.8	<i>Sequential Minimal Optimization (SMO)</i> .....	32
2.8.1	<i>Simplified Sequential Minimal Optimization (SSMO)</i> .....	35
2.9	Rata-rata ( <i>Mean</i> ).....	39
2.10	Evaluasi.....	40
<b>BAB III METODOLOGI PENELITIAN</b> .....		<b>41</b>
3.1	Studi Literatur.....	41
3.2	Pengumpulan Data.....	42
3.3	Analisis Kebutuhan.....	42
3.4	Perancangan Sistem.....	43
3.5	Implementasi Sistem.....	44
3.6	Pengujian dan Analisis.....	45
3.7	Kesimpulan.....	45
<b>BAB IV PERANCANGAN</b> .....		<b>47</b>
4.1	Analisis Kebutuhan Perangkat Lunak.....	48
4.1.1	Identifikasi Pengguna.....	48
4.1.2	Daftar Kebutuhan Sistem.....	49
4.2	Perancangan SPK.....	49
4.2.1	Subsistem Manajemen Berbasis Pengetahuan.....	51
4.2.2	Subsistem Manajemen Data.....	57
4.2.3	Subsistem Manajemen Model.....	70
4.2.4	Subsistem Antarmuka.....	122
<b>BAB V IMPLEMENTASI</b> .....		<b>128</b>
5.1	Spesifikasi Sistem.....	129
5.1.1	Spesifikasi Perangkat Keras.....	129
5.1.2	Spesifikasi Perangkat Lunak.....	129
5.2	Implementasi Algoritma.....	129
5.2.1	Algoritma <i>Simplified SMO</i> .....	130
5.2.2	Algoritma <i>One-Against-All</i> .....	135

5.3	Implementasi Antarmuka .....	135
5.3.1	Halaman Login .....	135
5.3.2	Halaman Petugas BPTU .....	136
5.3.3	Halaman Olah Data Pengguna .....	138
5.3.4	Halaman Peternak .....	139
5.3.5	Halaman Pengujian .....	139
<b>BAB VI PENGUJIAN DAN ANALISIS .....</b>		<b>140</b>
6.1	Pengujian Pengaruh Parameter .....	141
6.1.1	Pengujian Tingkat Akurasi Terhadap Pengaruh Ukuran <i>Cropping</i> dan Rasio Perbandingan Data Latih dan Data Uji .....	141
6.1.1.1	Skenario Pengujian 1 .....	141
6.1.1.2	Hasil Skenario Pengujian 1 .....	141
6.1.1.3	Analisis Hasil Skenario Pengujian 1 .....	144
6.1.2	Pengujian Tingkat Akurasi Terhadap Pengaruh Nilai Parameter C .....	144
6.1.2.1	Skenario Pengujian 2 .....	145
6.1.2.2	Hasil Skenario Pengujian 2 .....	145
6.1.2.3	Analisis Skenario Pengujian 2 .....	146
6.1.3	Pengujian Tingkat Akurasi Terhadap Pengaruh Nilai Parameter <i>Gamma</i> ( $\gamma$ ) .....	147
6.1.3.1	Skenario Pengujian 3 .....	147
6.1.3.2	Hasil Skenario Pengujian 3 .....	147
6.1.3.3	Analisis Skenario Pengujian 3 .....	149
6.1.4	Pengujian Tingkat Akurasi Terhadap Pengaruh Jumlah Iterasi .....	149
6.1.4.1	Skenario Pengujian 4 .....	150
6.1.4.2	Hasil Skenario Pengujian 4 .....	150
6.1.4.3	Analisis Skenario Pengujian 4 .....	151
6.2	Pengujian Akurasi .....	152
6.2.1	Pengujian Tingkat Akurasi Terhadap Nilai Klasifikasi Sistem Dan Pakar .....	152
6.2.1.1	Skenario Pengujian .....	152
6.2.1.2	Hasil Skenario Pengujian .....	152
6.2.1.3	Analisis Skenario Pengujian .....	154
<b>BAB VII PENUTUP .....</b>		<b>155</b>
7.1	Kesimpulan .....	155

7.2 Saran..... 156  
**DAFTAR PUSTAKA..... 157**



## DAFTAR GAMBAR

Gambar 2.1 Sapi Bali jantan dan betina dewasa.....	13
Gambar 2.2 Arsitektur Pemodelan SPK .....	18
Gambar 2.3 KoordinatCitra Digital .....	23
Gambar 2.4 Citra biner dan rerepresentasi citra biner .....	20
Gambar 2.5 Citra <i>grayscale</i> .....	20
Gambar 2.6 Citra warna.....	21
Gambar 2.7 <i>Hyperplane</i> yang memisahkan dua kelas .....	23
Gambar 2.8 Transformasi data ke <i>feature space</i> .....	26
Gambar 2.9 Contoh klasifikasi <i>One-Against-All</i> untuk 3 kelas.....	29
Gambar 2.10 Contoh klasifikasi dengan metode <i>One-Against-One</i> .....	30
Gambar 2.11 Contoh klasifikasi dengan metode DAGSVM.....	31
Gambar 2.12 <i>Pseudocode Simplified SMO</i> .....	38
Gambar 3.1 Diagram Alir Metodologi Penelitian .....	41
Gambar 3.2 Diagram Blok SPK Pemilihan Bibit Unggul Sapi Bali .....	43
Gambar 3.3 Aristektur SPK Pemilihan Bibit Unggul Sapi Bali .....	44
Gambar 4.1 Pohon Perancangan .....	47
Gambar 4.2 Aristektur SPK Pemilihan Bibit Unggul Sapi Bali .....	50
Gambar 4.3 Diagram Alir SPK Pemilihan Bibit Unggul Sapi Bali .....	51
Gambar 4.4 Sapi Bali Jantan Kualitas Baik .....	52
Gambar 4.5 Sapi Bali Betina Kualitas Baik .....	52
Gambar 4.6 Sapi Bali Jantan Kualitas Sedang .....	53
Gambar 4.7 Sapi Bali Betina Kualitas Sedang .....	53
Gambar 4.8 Sapi Bali Jantan Kualitas Buruk .....	54
Gambar 4.9 Sapi Bali Betina Kualitas Buruk .....	54
Gambar 4.10 Sapi Bali Betina .....	55
Gambar 4.11 Sapi Bali Betina Setelah Dilakukan Proses Perbesaran .....	55
Gambar 4.12 Diagram Konteks .....	57
Gambar 4.13 DFD Level 1 .....	58
Gambar 4.14 DFD Level 2 <i>Login</i> .....	59
Gambar 4.15 DFD Level 2 Olah Data Sapi Bali .....	59

Gambar 4.16 DFD Level 2 Olah Data Pengguna .....	60
Gambar 4.17 DFD Level 2 Olah Data Peternak .....	60
Gambar 4.18 DFD Level 2 Pengujian .....	61
Gambar 4.19 ERD SPK Pemilihan Bibit Unggul Sapi Bali .....	62
Gambar 4.20 PDM SPK Pemilihan Bibit Unggul Sapi Bali .....	64
Gambar 4.21 Diagram Alir Proses Tambah Data Sapi Bali .....	71
Gambar 4.22 <i>Pseudocode</i> Tambah Data Sapi Bali .....	71
Gambar 4.23 Diagram Alir Proses Ubah Data Sapi Bali .....	72
Gambar 4.24 <i>Pseudocode</i> Ubah Data Sapi Bali .....	73
Gambar 4.25 Diagram Alir Proses Hapus Data Sapi Bali .....	74
Gambar 4.26 <i>Pseudocode</i> Hapus Data Sapi Bali .....	74
Gambar 4.27 Diagram Alir Proses Tambah Data Pengguna .....	75
Gambar 4.28 <i>Pseudocode</i> Tambah Data Pengguna .....	75
Gambar 4.29 Diagram Alir Proses Ubah Data Pengguna .....	76
Gambar 4.30 <i>Pseudocode</i> Ubah Data Pengguna .....	77
Gambar 4.31 Diagram Alir Proses Hapus Data Pengguna .....	77
Gambar 4.32 <i>Pseudocode</i> Ubah Data Pengguna .....	78
Gambar 4.33 Diagram Alir Algoritma <i>Simplified SMO</i> .....	78
Gambar 4.34 Diagram Alir Proses <i>Cropping</i> .....	79
Gambar 4.35 <i>Pseudocode Cropping</i> Citra Sapi Bali .....	79
Gambar 4.36 Diagram Alir Ekstraksi Fitur .....	80
Gambar 4.37 <i>Pseudocode</i> Ekstraksi Fitur .....	81
Gambar 4.38 Diagram Alir Algoritma <i>Simplified SMO</i> .....	83
Gambar 4.39 <i>Pseudocode</i> Algoritma <i>Simplified SMO</i> .....	84
Gambar 4.40 Diagram Alir Perhitungan <i>Error</i> Pada Data ke- <i>i</i> .....	86
Gambar 4.41 <i>Pseudocode</i> Perhitungan <i>Error</i> Pada Data ke- <i>i</i> .....	87
Gambar 4.42 Diagram Alir Pengecekan kondisi KKT .....	88
Gambar 4.43 <i>Pseudocode</i> Pengecekan kondisi KKT .....	88
Gambar 4.44 Diagram Alir Mencari Data Pembanding Dan Nilai <i>Error</i> Data ke- <i>j</i> .....	89
Gambar 4.45 <i>Pseudocode</i> Data Pembanding Dan Nilai <i>Error</i> Pada Data ke- <i>j</i> .....	90
Gambar 4.46 Diagram Alir Men- <i>set</i> Nilai Alpha .....	91
Gambar 4.47 <i>Pseudocode</i> Men- <i>set</i> Nilai Alpha .....	91

Gambar 4.48 Diagram Alir Menghitung Segmen Garis Diagonal .....	92
Gambar 4.49 <i>Pseudocode</i> Menghitung Segmen Garis Diagonal .....	93
Gambar 4.50 Diagram Alir Mengecek Nilai L dan H .....	94
Gambar 4.51 <i>Pseudocode</i> Mengecek Nilai L dan H .....	95
Gambar 4.52 Diagram Alir Menghitung Nilai <i>Eta</i> .....	96
Gambar 4.53 <i>Pseudocode</i> Menghitung Nilai <i>Eta</i> .....	96
Gambar 4.54 Diagram Alir Menghitung Nilai Alpha Baru Pada Data ke- <i>j</i> .....	97
Gambar 4.55 <i>Pseudocode</i> Menghitung Nilai Alpha Baru Pada Data ke- <i>j</i> .....	98
Gambar 4.56 Diagram Alir Menentukan Batasan Maksimum Nilai Alpha .....	99
Gambar 4.57 <i>Pseudocode</i> Menentukan Batasan Maksimum Nilai Alpha .....	99
Gambar 4.58 Diagram Alir Mengecek Nilai Absolut Alpha Baru Data ke- <i>j</i> .....	100
Gambar 4.59 <i>Pseudocode</i> Mengecek Nilai Absolut Alpha Baru Data ke- <i>j</i> .....	100
Gambar 4.60 Diagram Alir Menghitung Nilai Alpha Data ke- <i>i</i> .....	101
Gambar 4.61 <i>Pseudocode</i> Menghitung Nilai Alpha Data ke- <i>i</i> .....	102
Gambar 4.62 Diagram Alir Menghitung Nilai Bias .....	103
Gambar 4.63 <i>Pseudocode</i> Menghitung Nilai Bias.....	103
Gambar 4.64 Diagram Alir Metode <i>One-Against-All</i> .....	119
Gambar 4.65 <i>Pseudocode</i> Metode <i>One-Against-All</i> .....	120
Gambar 4.66 Diagram Alir Konsep Dasar Metode <i>One-Against-All</i> .....	121
Gambar 4.67 Antarmuka <i>Login</i> .....	122
Gambar 4.68 Antarmuka Petugas BPTU .....	123
Gambar 4.69 Antarmuka Olah Data Pengguna .....	124
Gambar 4.70 Antarmuka Peternak .....	125
Gambar 4.71 Antarmuka Pengujian .....	126
Gambar 5.1 Pohon Implementasi.....	128
Gambar 5.2 Halaman Login.....	136
Gambar 5.3 Halaman Petugas BPTU .....	136
Gambar 5.4 Halaman Tambah Data Sapi Bali.....	137
Gambar 5.5 Halaman Ubah Data Sapi Bali .....	137
Gambar 5.6 Halaman Lihat Detail Data Sapi Bali.....	138
Gambar 5.7 Halaman Olah Data Pengguna .....	138
Gambar 5.8 Halaman Peternak .....	139

Gambar 5.9 Halaman Pengujian ..... 139

Gambar 6.1 Pohon Pengujian Dan Analisis..... 140

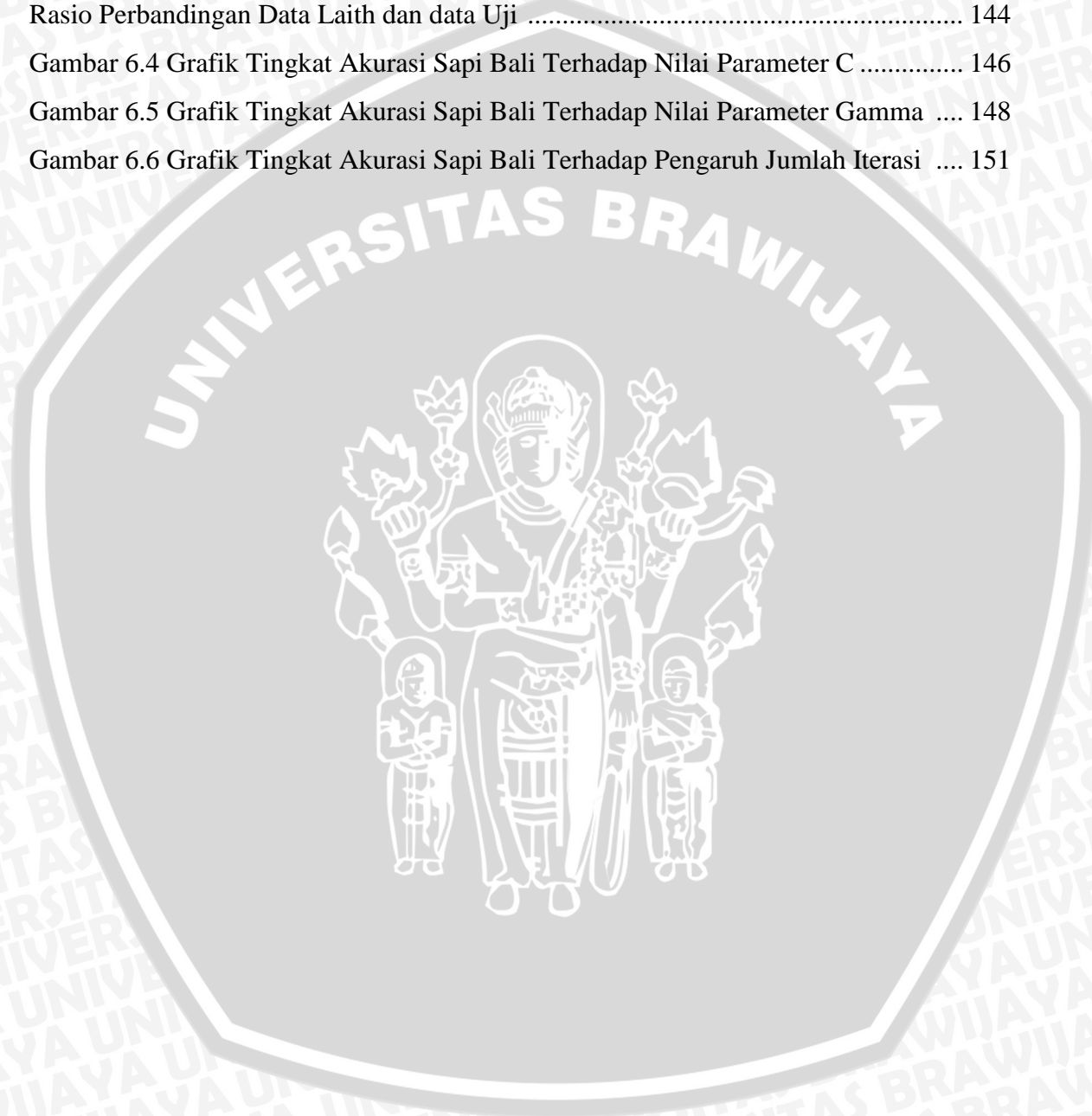
Gambar 6.2 Grafik Tingkat Akurasi Sapi Bali Jantan Terhadap Ukuran *Cropping* dan Rasio Perbandingan Data Laith dan data Uji ..... 142

Gambar 6.3 Grafik Tingkat Akurasi Sapi Bali Betina Terhadap Ukuran *Cropping* dan Rasio Perbandingan Data Laith dan data Uji ..... 144

Gambar 6.4 Grafik Tingkat Akurasi Sapi Bali Terhadap Nilai Parameter C ..... 146

Gambar 6.5 Grafik Tingkat Akurasi Sapi Bali Terhadap Nilai Parameter Gamma .... 148

Gambar 6.6 Grafik Tingkat Akurasi Sapi Bali Terhadap Pengaruh Jumlah Iterasi .... 151





## DAFTAR TABEL

Tabel 2.1 Kajian Pustaka .....	7
Tabel 2.2 Standar Penilaian Sapi Bali Jantan Sapihan .....	14
Tabel 2.3 Standar Penilaian Sapi Bali Jantan Umur 1 Tahun .....	14
Tabel 2.4 Standar Penilaian Sapi Bali Jantan Umur 2 Tahun .....	15
Tabel 2.5 Standar Penilaian Sapi Bali Betina Sapihan .....	15
Tabel 2.6 Standar Penilaian Sapi Bali Betina Umur 1 Tahun .....	16
Tabel 2.7 Standar Penilaian Sapi Bali Betina Umur 1.5 Tahun .....	15
Tabel 2.8 Metode <i>One-Against-All</i> dengan 4 Kelas .....	29
Tabel 2.9 Metode <i>One-Against-One</i> dengan 6 SVM Biner .....	30
Tabel 2.10 Metode DAGSVM dengan 6 SVM Biner .....	31
Tabel 2.11 Nilai RGB Pada Citra Berukuran 3x3 Pixel .....	39
Tabel 4.1 Identifikasi Pengguna .....	48
Tabel 4.2 Daftar Kebutuhan Fungsional .....	49
Tabel 4.3 Kriteria Ftur Warna Kulit Sapi Bali .....	56
Tabel 4.4 Struktur Tabel User .....	65
Tabel 4.5 Struktur Tabel Data Sapi Bali .....	65
Tabel 4.6 Struktur Tabel Crop Sapi Bali .....	65
Tabel 4.7 Struktur Tabel Info Crop Sapi Bali .....	66
Tabel 4.8 Struktur Tabel Data Rasio .....	66
Tabel 4.9 Struktur Tabel Data C .....	67
Tabel 4.10 Struktur Tabel Data Gamma .....	67
Tabel 4.11 Struktur Tabel Data Iterasi .....	68
Tabel 4.12 Struktur Tabel Da Rasio .....	68
Tabel 4.13 Struktur Tabel Da C .....	69
Tabel 4.14 Struktur Tabel Da Gamma .....	69
Tabel 4.15 Struktur Tabel Da Iterasi .....	69
Tabel 4.16 Data Pelatihan Pertama .....	85
Tabel 4.17 Hasil Perhitungan dengan Kernel RBF .....	86
Tabel 4.18 Hasil <i>Error</i> Pada Data ke- <i>i</i> .....	87
Tabel 4.19 Kondisi KKT .....	88

Tabel 4.20 <i>Error Data</i> Pemanding ( $E_j$ ) .....	90
Tabel 4.21 Nilai Alpha data $i$ dan $j$ .....	91
Tabel 4.22 Nilai L dan H (target sama) .....	93
Tabel 4.23 Nilai L dan H (target tidak sama) .....	93
Tabel 4.24 Nilai L dan H (Kondisi) .....	95
Tabel 4.25 Hasil Perhitungan <i>Eta</i> .....	96
Tabel 4.26 Nilai Alpha Baru data ke- $j$ .....	98
Tabel 4.27 Nilai Alpha data $j$ pada segmen garis .....	99
Tabel 4.28 Kondisi Nilai Absolut .....	101
Tabel 4.29 Nilai Alpha Baru data ke- $i$ .....	102
Tabel 4.30 Nilai $b_1$ dan $b_2$ .....	104
Tabel 4.31 Nilai bias akhir .....	105
Tabel 4.32 Nilai Alpha dan Bias Pada Pelatihan Pertama .....	105
Tabel 4.33 Data Pelatihan Kedua .....	106
Tabel 4.34 Hasil Error pada data ke- $i$ .....	107
Tabel 4.35 Kondisi KKT .....	108
Tabel 4.36 <i>Error Data</i> pemanding .....	108
Tabel 4.37 Nilai Alpha data $i$ dan $j$ .....	109
Tabel 4.38 Nilai L dan H (target sama) .....	110
Tabel 4.39 Nilai L dan H (target tidak sama) .....	111
Tabel 4.40 Nilai L dan H (Kondisi) .....	111
Tabel 4.41 Hasil Perhitungan <i>Eta</i> .....	112
Tabel 4.42 Nilai Alpha Baru data ke- $j$ .....	113
Tabel 4.43 Nilai Alpha data $j$ pada segmen garis .....	114
Tabel 4.44 Kondisi Nilai Absolut .....	115
Tabel 4.45 Nilai Alpha Baru data ke- $i$ .....	115
Tabel 4.46 Nilai $b_1$ dan $b_2$ .....	117
Tabel 4.47 Nilai bias akhir .....	118
Tabel 4.48 Nilai Alpha dan Bias Pada Pelatihan Pertama .....	118
Tabel 4.49 Data Uji .....	120
Tabel 4.50 Hasil Perhitungan Kernel RBF Data Uji dan Data Latih .....	120
Tabel 5.1 Spesifikasi Perangkat Keras .....	129



Tabel 5.2 Spesifikasi Perangkat Lunak.....	129
Tabel 5.3 Implementasi Algoritma <i>Simplified SMO</i> .....	130
Tabel 5.4 Implementasi Algoritma <i>One-Against-All</i> .....	135
Tabel 6.1 Perbandingan Jumlah Data Latih dan Data Uji .....	141
Tabel 6.2 Tingkat Akurasi Sapi Bali Jantan dengan Ukuran Cropping 32 px (%).....	141
Tabel 6.3 Tingkat Akurasi Sapi Bali Jantan dengan Ukuran Cropping 64 px (%).....	142
Tabel 6.4 Tingkat Akurasi Sapi Bali Jantan dengan Ukuran Cropping 128 px (%).....	142
Tabel 6.5 Tingkat Akurasi Sapi Bali Betina dengan Ukuran Cropping 32 px (%) .....	143
Tabel 6.6 Tingkat Akurasi Sapi Bali Betina dengan Ukuran Cropping 64 px (%) .....	143
Tabel 6.7 Tingkat Akurasi Sapi Bali Betina dengan Ukuran Cropping 128 px (%) ....	143
Tabel 6.8 Tingkat Akurasi Sapi Bali Jantan Terhadap Pengaruh Nilai C (%) .....	145
Tabel 6.9 Tingkat Akurasi Sapi Bali Betina Terhadap Pengaruh Nilai C (%).....	145
Tabel 6.10 Tingkat Akurasi Sapi Bali Jantan Terhadap Pengaruh Nilai <i>Gamma</i> (%) .	148
Tabel 6.11 Tingkat Akurasi Sapi Bali Betina Terhadap Pengaruh Nilai <i>Gamma</i> (%) .	148
Tabel 6.12 Tingkat Akurasi Sapi Bali Jantan Terhadap Pengaruh Jumlah Iterasi (%)	150
Tabel 6.13 Tingkat Akurasi Sapi Bali Betina Terhadap Pengaruh Jumlah Iterasi (%)	150
Tabel 6.14 Tingkat Akurasi Sapi Bali Jantan .....	153
Tabel 6.15 Tingkat Akurasi Sapi Bali Betina.....	153



## DAFTAR LAPMPIRAN

Lampiran 1 Data Gambar Sapi Bali Validasi .....	L1
Lampiran 2 Perbesaran Gambar Sapi Bali dengan <i>Photoshop</i> .....	L2
Lampiran 3 Dataset Sapi Bali .....	L3
Lampiran 4 <i>Cropping</i> Sapi Bali.....	L4
Lampiran 5 Representasi <i>Cropping</i> Sapi Bali .....	L5
Lampiran 6 Data Pengujian Sapi Bali.....	L6



## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Sapi Bali (*Bos sondaicus*) merupakan sapi asli dan murni dari Indonesia yang dihasilkan dari domestikasi banteng liar [8:15]. Sapi Bali merupakan hewan budidaya yang memiliki harga cukup mahal dan dagingnya memiliki kontribusi besar dalam memenuhi kebutuhan protein hewani bagi masyarakat Indonesia. Penilaian dan pemilihan bibit unggul Sapi Bali adalah dengan melihat penampilannya (*performance*). Faktor-faktor yang terdapat pada Sapi Bali dalam pemilihan bibit unggul antara lain berat badan, tempramen, warna kulit, cacat tubuh, *scrotum*, ambing dan bentuk perototan. 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 [18:75-76]. 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 tersebut terjadi dikarenakan peternak melihat secara langsung pola warna kulit dari Sapi Bali yang akan dijadikan bibit unggul mengingat terbatasnya kemampuan mata manusia dalam klasifikasi dengan jumlah Sapi Bali yang banyak di BPTU. Pemilihan bibit unggul Sapi Bali akan diklasifikasikan ke dalam tiga kelas antara lain Baik (Bibit Unggul), Sedang, dan Buruk (Penyimpangan)[1:3]. 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. Penelitian ini akan diterapkan pada sistem pendukung keputusan yang bertujuan untuk membantu peternak dalam proses klasifikasi kualitas dan pengambilan keputusan untuk pemilihan bibit unggul Sapi Bali berdasarkan warna kulit. Fitur yang digunakan dalam proses pengklasifikasian merupakan komponen warna dari warna kulit Sapi Bali yaitu *red*, *green* dan *blue* (RGB)[15:37].

Penelitian yang berkaitan dengan penggunaan citra digital sebagai objek utama dan fitur RGB dilakukan oleh Eliyani, Tulus dan Fahmi yang berjudul “*Pengenalan Tingkat Kematangan Pepaya Paya Rabo Menggunakan Pengolahan Citra Digital Berdasarkan Warna RGB dengan K-Means Clustering*”. Penelitian ini menggunakan metode *K-Means Clustering* yang digunakan untuk mengelompokan citra buah pepaya paya rabo berdasarkan tingkat kematangannya menggunakan fitur *red*, *green*, dan *blue* (RGB). Tingkat kematangan yang dikelompokan antara lain masak muda, masak mengkal dan masak penuh [6:93-97]. Penelitian selanjutnya adalah penelitian yang dilakukan oleh Deepti Vadicherla dan Sheetal Sonawane 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 [21:19-26].

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*. Algoritma *Simplified SMO* merupakan pengembangan dari algoritma SMO, keunggulan algoritma ini dapat mempercepat optimasi nilai  $\alpha$  dan *bias* pada dataset yang memiliki jumlah yang besar [16:1]. Berdasarkan permasalahan yang telah dipaparkan, penulis mengusulkan skripsi dengan judul “**Sistem Pendukung Keputusan Pemilihan Bibit Unggul Sapi Bali Menggunakan Algoritma *Simplified Sequential Minimal Optimization* (SSMO) pada *Support Vector Machine* (SVM)**”.

## 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana merancang dan membangun suatu sistem pendukung keputusan untuk pemilihan bibit unggul Sapi Bali menggunakan algoritma *Simplified SMO* berdasarkan warna kulit ?

2. Bagaimana mengimplementasi algoritma *Simplified SMO* untuk pemilihan bibit unggul Sapi Bali?
3. Bagaimana mengevaluasi tingkat akurasi dari pemilihan bibit unggul Sapi Bali menggunakan algoritma *Simplified SMO*?

### 1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Data input berupa citra digital berformat (\*.jpg).
2. Pengambilan citra Sapi dilakukan di Balai Pembibitan Ternak Unggul Sapi Bali (BPTU) yang berada di Jembrana, Bali.
3. Algoritma yang digunakan adalah algoritma *Simplified Sequential Minimal Optimization* (SSMO) dengan Kernel *Radial Basis Function* (RBF).
4. Ukuran *cropping* yang digunakan yaitu  $32 \times 32 \text{ px}$ ,  $64 \times 64 \text{ px}$ , dan  $128 \times 128 \text{ px}$ .
5. Fitur yang digunakan nilai rata-rata *red*, rata-rata *green* dan rata-rata *blue* (RGB) dari *cropping* citra Sapi Bali pada bagian tubuh samping dari Sapi Bali.
6. Hasil klasifikasi yang diberikan yaitu Baik, Sedang dan Buruk.
7. Faktor-faktor yang digunakan dalam penelitian ini hanya digunakan faktor warna kulit.
8. Sapi Bali yang digunakan sebagai objek penelitian yaitu fokus pada Sapi Bali dewasa (Sapi Bali Jantan berumur 2 tahun ke atas sedangkan Sapi Bali Betina berumur 1,5 tahun ke atas).
9. Variabel umur pada Sapi Bali tidak digunakan dalam proses klasifikasi.
10. Aplikasi yang dibangun menggunakan bahasa *Java* dan *MySql* sebagai DBMS.

### 1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini adalah merancang sebuah sistem pendukung keputusan pemilihan bibit unggul Sapi Bali berdasarkan warna kulit dengan menerapkan algoritma *Simplified SMO* serta mengukur tingkat akurasi implementasi algoritma tersebut.

### 1.5 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat bagi penulis antara lain:

1. Mendapatkan pengetahuan terkait algoritma-algoritma yang terdapat dalam klasifikasi.
2. Menambah pengetahuan dalam menerapkan algoritma *Simplified SMO* pada SPK Pemilihan Bibit Unggul Sapi Bali Berdasarkan Warna Kulit.

Sedangkan manfaat bagi pihak ketiga adalah:

1. Mengurangi tingkat kesalahan dalam pemilihan bibit unggul Sapi Bali
2. Memudahkan para peternak dalam pemilihan bibit unggul Sapi Bali.
3. Mempercepat proses klasifikasi dan pemilihan bibit unggul Sapi Bali.
4. Tersedianya teknologi informasi untuk klasifikasi dan pemilihan bibit unggul Sapi Bali.

### 1.6 Sistematika Penulisan

Penelitian ini bertujuan untuk meneliti klasifikasi Sapi Bali berdasarkan fitur warna kulit. Pada penelitian ini, peneliti akan memaparkan sistematika penulisan sebagai berikut:

#### **BAB I           Pendahuluan**

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika penulisan.

#### **BAB II           Tinjauan Pustaka**

Menguraikan tentang kajian pustaka serta teori penunjang yang berhubungan dengan Sapi Bali, SPK, citra digital, SMO, *Simplified SMO* serta SVM.

#### **BAB III         Metodologi Penelitian**

Membahas mengenai langkah-langkah kerja yang dilakukan dalam penelitian ini yang terdiri dari studi literatur, analisis kebutuhan, perancangan sistem, implementasi sistem, serta pengujian dan analisis.

#### **BAB IV         Perancangan**

Membahas analisis kebutuhan dan perancangan *user interface* untuk pengembangan perangkat lunak.

#### **BAB V           Implementasi**

Membahas tentang implementasi algoritma *Simplified SMO* pada studi kasus pemilihan bibit unggul Sapi Bali.

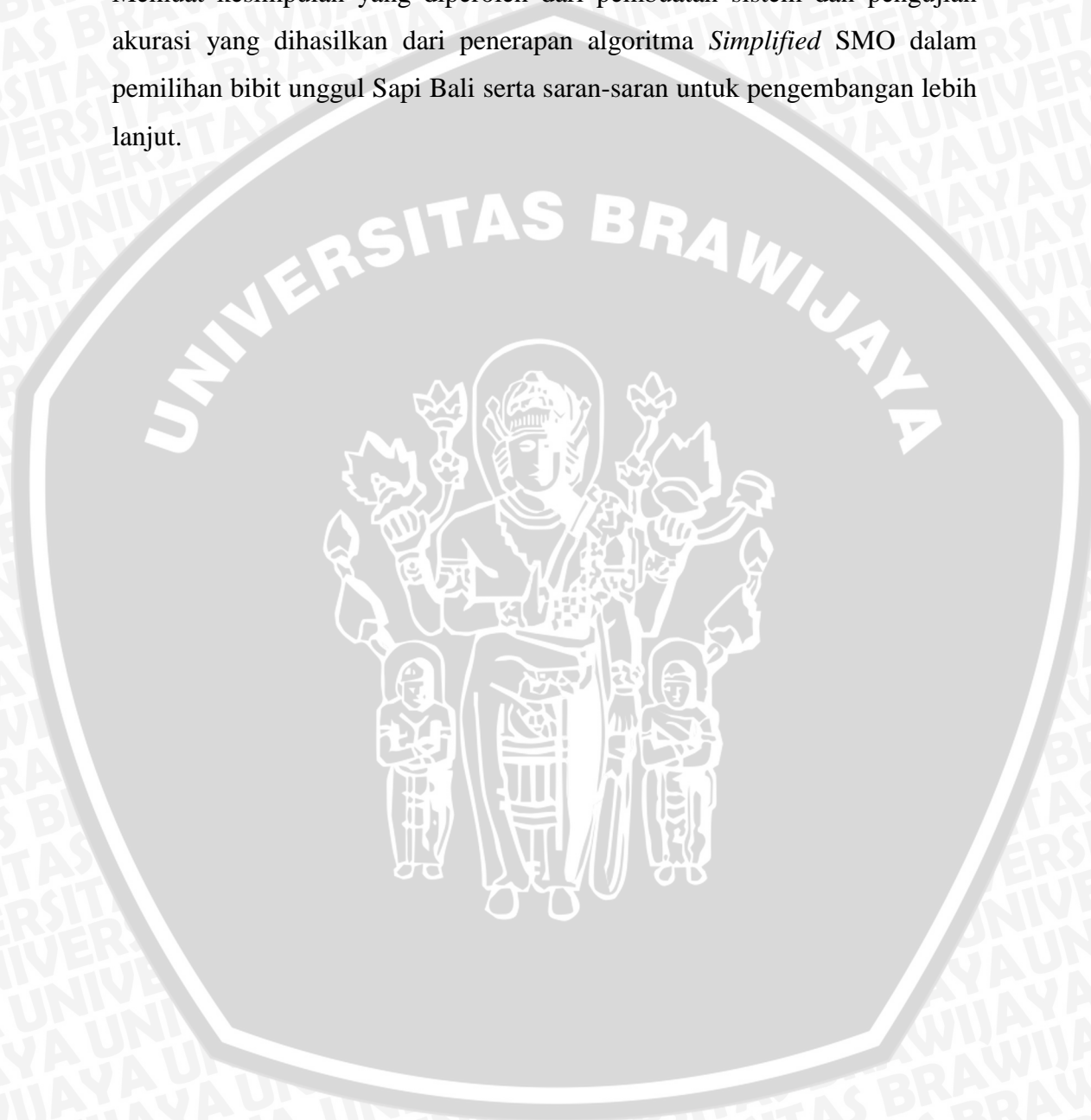


## **BAB VI            Pengujian dan Analisis**

Membahas hasil pengujian dan analisis terhadap sistem yang telah dibangun dengan menggunakan algoritma *Simplified SMO*.

## **BAB VII            Penutup**

Memuat kesimpulan yang diperoleh dari pembuatan sistem dan pengujian akurasi yang dihasilkan dari penerapan algoritma *Simplified SMO* dalam pemilihan bibit unggul Sapi Bali serta saran-saran untuk pengembangan lebih lanjut.



## BAB II

### TINJAUAN PUSTAKA

Bab ini terdiri dari kajian pustaka dan dasar teori yang berhubungan dengan penelitian ini. Kajian pustakan membahas penelitian sebelumnya dan penggunaan metode pada kasus yang berbeda. Dasar teori membahas teori penunjang yang berkaitan dengan penelitian diantaranya Sapi Bali, sistem pendukung keputusan, citra digital, SVM dan algoritma *Simplified Sequential Minimal Optimization* (SSMO).

#### 2.1 Kajian Pustaka

Kajian pustaka pada penelitian ini akan membahas beberapa penelitian sebelumnya yang relevan dengan judul skripsi yang akan dibuat. Beberapa penelitian tersebut dilakukan menggunakan algoritma SMO, SVM serta pengenalan pola warna pada citra digital.

Terdapat lima penelitian yang digunakan oleh penulis sebagai referensi dalam penelitian yang akan dilakukan. Penelitian pertama dilakukan oleh Eliyani, Tulus dan Fahmi. Penelitian ini berfokus pada objek citra digital yaitu citra buah pepaya paya rabo yang akan diketahui tingkat kematangannya. Penelitian kedua dilakukan oleh Siti Munawaroh dan Felix Andreas Sutanto. Objek pada penelitian ini adalah uang kertas yang nantinya akan diidentifikasi nominalnya.

Penelitian ketiga yang dilakukan oleh Dewi Y.Liliana, Agung Hardianto dan M. Ridok. Objek pada penelitian ini adalah dokumen berita berbahasa Indonesia yang akan diklasifikasikan berdasarkan kategori yang telah ditentukan sebelumnya. Penelitian keempat dilakukan oleh R. Guruvasuki dan A. Josephine Puspha Arasi M.E. Penelitian ini berfokus pada objek citra digital yaitu citra MRI otak manusia. Penelitian kelima dilakukan oleh Deepti Vadicherla dan Sheetal Sonawane. Objek pada penelitian ini berupa data penyakit jantung pasien. Uraian dan penjelasan terhadap kelima penelitian tersebut ditunjukkan pada tabel 2.1.

Tabel 2.1 Kajian Pustaka

No	Tahun/Judul	Obyek (Input)	Metode (Proses)	Hasil (Output)
1	[2010]Pengolahan Citra Digital untuk Identifikasi Uang Kertas [15:34-40].	Identifikasi Uang Kertas <ul style="list-style-type: none"> <li>➤ Data citra uang kertas dengan nominal Rp.1000,-,Rp.5000,-,Rp.10000,-,Rp.20000,-,Rp.50000,-,Rp.100000,-.</li> <li>➤ Nilai rata-rata <i>red</i>, <i>green</i> dan <i>blue</i> (RGB).</li> </ul>	Pengolahan Citra Digital (perbaikan citra, segmentasi dan pencocokan) Langkah-langkah: <ol style="list-style-type: none"> <li>1. Melakukan pengambilan citra uang kertas dengan bantuan scanner.</li> <li>2. Melakukan perbaikan citra(kontras dan <i>brightness</i>).</li> <li>3. Melakukan segmentasi melalui <i>cropping</i> dengan ukuran tertentu.</li> <li>4. Melakukan ekstraksi ciri (fitur) untuk mendapatkan rata-rata RGB.</li> <li>5. Melakukan pencocokan data antara query dengan pola yang terdapat pada database.</li> </ol>	<ul style="list-style-type: none"> <li>➤ Output dari penelitian ini pengenalan terhadap uang kertas dengan nominal 1000, 5000, 10000, 20000, 50000, dan 100000.</li> <li>➤ Akurasi yang didapatkan dari penelitian ini adalah 86% dengan menggunakan 6 data training dan 9 data uji.</li> </ul>
2	[2011]Indonesian News Classification using Support Vector Machine [10:694-697].	Klasifikasi Berita Berbahasa Indonesia. <ul style="list-style-type: none"> <li>➤ Berita berbahasa Indonesia yang diambil dari situs <a href="http://www.kompas.com">www.kompas.com</a>.</li> <li>➤ Nilai TF-IDF yang dihasilkan dari tiap kata pada berita.</li> </ul>	Metode Support Vector Machine Langkah-langkah: <ol style="list-style-type: none"> <li>1. Melakukan input data <i>training</i> ke dalam sistem.</li> <li>2. Melakukan proses <i>tokenizing</i> terhadap data <i>training</i>.</li> <li>3. Melakukan <i>stopword filtering</i> terhadap data <i>training</i>.</li> <li>4. Melakukan <i>stemming</i> terhadap data <i>training</i>.</li> <li>5. Menghitung frekuensi kata pada data <i>training</i>.</li> <li>6. Mendapatkan nilai Tf-Idf yang akan dijadikan fitur.</li> <li>7. Melakukan normalisasi pada fitur yang telah didapatkan.</li> </ol>	<ul style="list-style-type: none"> <li>➤ Kategori dari hasil output penelitin ini adalah berita nasional, internasional, bisnis dan keuangan, dan olahraga.</li> <li>➤ Rata-rata yang dihasilkan dari aplikasi ini sebesar 85%.</li> </ul>

<p>3</p>	<p>[2013]Pengenalan Tingkat Kematangan Buah Pepaya Paya Rabo Menggunakan Pengolahan Citra Berdasarkan Warna RGB dengan <i>K-Means Clustering</i>[6:93-97].</p>	<p>Pengenalan Tingkat Kematangan Buah Pepaya</p> <ul style="list-style-type: none"> <li>➤ Data citra pepaya paya rabo</li> <li>➤ Nilai komponen warna <i>red</i>, <i>green</i> dan <i>blue</i> (RGB)</li> </ul>	<p>Metode <i>K-Means Clustering</i> Langkah-langkah:</p> <ol style="list-style-type: none"> <li>1. Melakukan pengambilan data berupa citra buah pepaya paya rabo.</li> <li>2. Melakukan normalisasi ukuran pada citra buah pepaya.</li> <li>3. Melakukan proses pengambilan nilai RGB.</li> <li>4. Melakukan pengukuran jarak menggunakan metode <i>euclidean distance</i>.</li> <li>5. Melakukan pengklasteran (kelompok) citra pepaya dengan menentukan banyak klastering yang diinginkan.</li> </ol>	<ul style="list-style-type: none"> <li>➤ Output dari penelitian ini berupa tingkat kematangan dari pepaya paya rabo yaitu masak muda, masak mengkal, dan masak penuh.</li> <li>➤ Didapatkan akurasi sebesar 60% untuk masak muda, 90% masak mengkal dan 100% masak penuh dengan menggunakan 30 data uji dan 15 data training.</li> </ul>
<p>4</p>	<p>[2013]MRI Brain Image Retireval Using Multi Support Vector Machine Classifier [9:29-36].</p>	<p>Klasifikasi Citra MRI Otak.</p> <ul style="list-style-type: none"> <li>➤ Data citra MRI otak manusia.</li> </ul>	<p>Metode Support Vector Machine Langkah-langkah:</p> <ol style="list-style-type: none"> <li>1. Melakukan <i>preprocessing</i>.</li> <li>2. Melakukan <i>low pass filtering</i>.</li> <li>3. Melakukan <i>median filtering</i>.</li> <li>4. Melakukan ekstraksi fitur dengan menggunakan algoritma <i>Gray Level Co-occurrence Matrix</i> (GLCM).</li> </ol>	<ul style="list-style-type: none"> <li>➤ Output dari penelitian ini adalah <i>normal</i>, <i>benign</i> dan <i>malignant</i>.</li> </ul>
<p>5</p>	<p>[2013]Decision Support System For Heart Disease Based On Sequential Minimal Optimization in Support Vector Machine [21:19-26]</p>	<p>Klasifikasi Diagnosa Penyakit Jantung.</p> <ul style="list-style-type: none"> <li>➤ Data set yang diambil dari UCI learning data set repository of Cleveland.</li> </ul>	<p>Algoritma <i>Sequential Minimal Optimization</i> (SMO) Langkah-langkah:</p> <ol style="list-style-type: none"> <li>1. Mendapatkan dataset berupa 14 fitur dari UCI</li> <li>2. Melakukan <i>learning</i> pada data latih untuk mendapatkan optimasi dari nilai alpha, bias dan <i>error-cache</i>.</li> <li>3. Melakukan klasifikasi dari data uji.</li> </ol>	<ul style="list-style-type: none"> <li>➤ Output dari penelitian ini berupa dua kelas yaitu ada penyakit jantung dan tidak ada penyakit jantung.</li> </ul>

<p>6</p>	<p>[2014]Sistem Pendukung Keputusan Pemilihan Bibit Unggul Sapi Bali Menggunakan Algoritma <i>Simplified Sequential Minimal Optimization</i> (SSMO) Pada <i>Support Vector Machine</i> (SVM).</p>	<p>Klasifikasi Bibit Unggul Sapi Bali</p> <ul style="list-style-type: none"> <li>➤ Data citra Sapi Bali yang diambil di BPTU Sapi Bali, Jembrana.</li> <li>➤ Nilai rata-rata dari <i>red</i>, <i>green</i> dan <i>blue</i> (RGB).</li> </ul>	<p>Algoritma <i>Simplified Sequential Minimal Optimization</i> (SMO)</p> <p>Langkah-langkah:</p> <ol style="list-style-type: none"> <li>1. Memasukkan citra Sapi Bali ke dalam sistem.</li> <li>2. Meng-<i>cropp</i> bagian samping pada Sapi Bali dengan ukuran tertentu.</li> <li>3. Melakukan proses untuk mendapatkan nilai rata-rata <i>red</i>, <i>green</i> dan <i>blue</i> (RGB) dari citra Sapi Bali yang telah di <i>cropp</i>.</li> <li>4. Melakukan pelatihan pada data latih dengan menggunakan algoritma <i>Simplified SMO</i> untuk mendapatkan optimasi dari nilai alpha dan bias.</li> <li>5. Melakukan proses pengujian terhadap data yang belum diketahui tipe kelasnya.</li> <li>6. Menggunakan metode <i>one-against-all</i> untuk menentukan kelas pada data uji.</li> </ol>	<ul style="list-style-type: none"> <li>➤ Output dari penelitian ini berupa tiga kelas yaitu Baik (Bibit Unggul), Sedang dan Buruk (Penyimpangan).</li> </ul>
----------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------

Sumber: [[6:93-97][15:34-40][10:694-697][9:29-36][21:19-26]]

Penelitian ketiga adalah penelitian yang dilakukan oleh Eliyani,dkk. Penelitian yang berjudul *Pengenalan Tingkat Kematangan Buah Pepaya Paya Rabo Menggunakan Pengolahan Citra Berdasarkan Warna RGB dengan K-Means Clustering* bertujuan untuk mengelompokkan citra buah pepaya paya rabo berdasarkan tingkat kematangannya. Fitur utama dalam penelitian ini adalah nilai rata-rata *red*, *green* dan *blue* (RGB) yang didapatkan dari citra pepaya paya rabo. Proses pengklasteran yang dilakukan dengan menggunakan metode *K-Means Clustering* melalui tiga tahap. Tahap pertama menetapkan suatu vektor matriks awal yang didapatkan dari fitur RGB. Tahap kedua mengklasifikasikan hasil dari pengukuran jarak terdekat menggunakan metode dari setiap data ke kluster terdekat. Tahap ketiga pengelompokan kedalam suatu kluster berdasarkan nilai dari rata-rata RGB [6:93-97].

Penelitian pertama adalah penelitian yang dilakukan oleh Siti Muwaroh,dkk. Penelitian yang berjudul *Pengolahan Citra Digital Untuk Identifikasi Uang Kertas*. Penelitian ini bertujuan untuk mengidentifikasi uang kertas berdasarkan warna yang dimilikinya. Pada proses identifikasi terhadap uang kertas ini dilakukan melalui beberapa tahap yaitu tahap *preprocessing*, tahap ekstraksi fitur dan tahap pencocokan (*matching*). Tahap *preprocessing* merupakan suatu tahap yang bertujuan untuk mendapatkan suatu informasi yang dibutuhkan dalam proses pencocokan. Tahap *preprocessing* ini meliputi perbaikan citra, segmentasi dan cropping. Tahap ekstraksi fitur bertujuan untuk mendapatkan fitur dari citra uang kertas. Fitur yang didapatkan merupakan nilai *red*, *green* dan *blue* (RGB). Tahap pencocokan merupakan tahap terakhir dalam penelitian ini. Tahap ini akan memberikan hasil berupa sebuah informasi berdasarkan query yang diinputkan oleh *user* [15:34-40].

Penelitian kedua adalah penelitian yang dilakukan Agung Hardianto,dkk. Penelitian yang berjudul *Indonesian News Classification using Support Vector Machine* bertujuan untuk mengklasifikasikan berita berbahasa Indonesia ke dalam empat kelas yaitu berita olahraga, bisnis dan keuangan, nasional dan internasional. Proses klasifikasi ini dibagi menjadi tiga tahap yaitu tahap *preprocessing*, tahap pelatihan dan tahap klasifikasi. Pada tahap *preprocessing* meliputi *tokenizing*, *stopword filtering*, *stemming*, menghitung frekuensi kata, menghitung fitur TF-IDF dan normalisasi. Tahap pelatihan menggunakan metode SVM yang bertujuan untuk

mencari support vector dan garis *hyperplane*. Proses ini memetakan data latih ke dalam dimensi yang lebih tinggi dengan menggunakan kernel *Radial Basis Function* (RBF). Tahap klasifikasi untuk menentukan data uji ke dalam kelas yang sebelumnya ditentukan [10:694-697].

Penelitian keempat adalah penelitian yang dilakukan oleh R. Guruvasuki, dkk. Penelitian ini berjudul *MRI Brain Image Retrieval Using Multi Support Vector Machine Classifier* bertujuan untuk mengklasifikasikan citra MRI otak manusia kedalam tiga kelas yaitu *normal*, *benign* dan *malignant*. Proses klasifikasi ini melalui tiga tahap yaitu *preprocessing*, *pelatihan* dan *klasifikasi*. Tahap *preprocessing* meliputi *low pass filtering*, *median filtering*, *ekstraksi fitur* menggunakan algoritma *Gray Level Co-occurrence Matrix* (GLCM). Tahap pelatihan menggunakan metode SVM berdasarkan nilai yang telah didapatkan dari hasil ekstraksi fitur. Tahap klasifikasi untuk menentukan citra uji MRI Otak ke dalam kelas *normal*, *benign* atau *malignant* [9:29-36].

Penelitian kelima adalah penelitian yang dilakukan oleh Deepti Vadicherla, dkk. Penelitian ini berjudul *Decision Support System For Heart Disease Based On Sequential Minimal Optimization In Support Vector Machine* bertujuan untuk mengklasifikasikan diagnosa medis untuk pasien yang memiliki penyakit jantung. Proses sistem keputusan ini dibagi menjadi dua bagian utama yaitu *training* dan *testing* (klasifikasi). Fitur yang digunakan dalam penelitian ini berjumlah 14 fitur yang berkaitan tentang penyakit jantung. Fitur-fitur yang digunakan antara lain umur, jenis kelamin, jenis sakit dada, tekanan darah, kolesterol, gula darah, hasil ECG, denyut jantung, latihan induksi angina, induksi depresi, *slope*, jumlah *vessel* yang berwarna, diagnosa penyakit jantung. Hasil klasifikasi dari penelitian ini yaitu ada penyakit jantung dan tidak ada penyakit jantung [21:19-26].

Penelitian ke-3 dan ke-1 menggunakan citra digital sebagai objek dalam penelitian. Fitur utama yang digunakan dalam penelitian tersebut adalah nilai *red*, *green* dan *blue* (RGB) yang didapatkan dari proses pengambilan fitur pada objek yang digunakan. Penggunaan fitur-fitur tersebut dapat menghasilkan rata-rata akurasi sebesar 80% pada masing-masing penelitian dengan menerapkan metode yang berbeda. Pada penelitian ke-2 dan ke-4 menggunakan menggunakan metode SVM. Penggunaan metode ini akan mengklasifikasi suatu objek ke dalam kelas yang telah

ditentukan sebelumnya dengan menggunakan fitur yang berbeda. Pada penelitian ke-5 menggunakan algoritma SMO pada SVM, penelitian ini mempunyai tujuan yang sama seperti penelitian ke-2 dan ke-4 yaitu pengklasifikasian.

Berdasarkan penelitian yang telah ada, maka penulis akan melakukan sebuah penelitian yang berkaitan dengan objek citra digital dan penggunaan algoritma *Simplified SMO*. Pada penelitian ini memiliki perbedaan dalam hal objek dan fitur dengan penelitian ke-2, ke-4 dan ke-5. Objek dan fitur yang digunakan dalam penelitian ini adalah citra Sapi Bali dan fitur *red*, *green* dan *blue* (RGB). Pada proses klasifikasi penelitian ini akan mengklasifikasikan citra Sapi Bali ke dalam tiga kelas yaitu baik, sedang dan buruk. Penelitian ini menggunakan metode *one-against-all* sebagai metode yang digunakan untuk proses pengklasifikasian *multi-class*. Dasar teori yang diperlukan untuk penelitian ini adalah Sapi Bali, SPK, citra digital, SVM dan algoritma *Simplified SMO*.

## 2.2 Sapi Bali

Indonesia merupakan negara yang memiliki kekayaan alam yang melimpah. Hal ini dapat dibuktikan bahwa sebagian besar mata pencaharian penduduk Indonesia di sektor pertanian yang mencakup peternakan, perikanan dan kehutanan. Peternakan adalah kegiatan mengembangbiakkan dan membudidayakan hewan ternak. Kegiatan di bidang peternakan dapat dibagi menjadi dua golongan, yaitu peternakan hewan besar dan peternakan hewan kecil.

Sapi merupakan salah satu dari ternak yang termasuk ke dalam golongan peternakan hewan besar. Indonesia memiliki beraneka ragam jenis sapi, salah satunya Sapi Bali. Sapi Bali (*Bos sondaicus*) merupakan Sapi asli dan murni dari Indonesia yang dihasilkan dari domestikasi banteng liar yang telah dilakukan sejak akhir abad ke 19 di Bali sehingga Sapi jenis ini dinamakan Sapi Bali.

Sapi Bali secara fisik tidak memiliki punuk, bertanduk, kepala agak pendek dengan dahi datar, kedua telinga tegak, bulu pada bagian dalam telinga putih, badan padat, dada dalam, kaki-kainya ramping agak pendek menyerupai kerbau, cermin hidung, kuku dan bulu ujung ekornya berwarna hitam.

Sapi Bali mempunyai klasifikasi taksonomi sebagai berikut [4:3] :

Phylum : *Chordata*

Sub Phylum : *Vertebrata*



Class	: <i>Mamalia</i>
Sub Class	: <i>Theria</i>
Infra Class	: <i>Eutheria</i>
Ordo	: <i>Artiodactyla</i>
Sub Ordo	: <i>Ruminantia</i>
Infra Ordo	: <i>Pecora</i>
Famili	: <i>Bovidae</i>
Genus	: <i>Bos (cattle)</i>
Group	: <i>Taurinae</i>
Spesies	: <i>Bos sondaicus</i> (banteng/Sapi Bali)



Gambar 2.1 Sapi Bali jantan dan betina dewasa  
Sumber : Observasi

Sapi Bali termasuk hewan *dimorphism-sex* yang artinya warna bulu akan berubah sesuai usia dan jenis kelaminnya. Pada sapihan (205 hari), bulu berwarna sawo matang sampai merah bata, namun warna bulu Sapi Bali jantan berubah menjadi coklat tua setelah mencapai dewasa dan menjadi hitam mulus pada umur 3 tahun. Perubahan warna disebabkan pengaruh hormon *testoteron* [8:19].

Sapi Bali memiliki beberapa keunggulan antara lain mempunyai fertilitas tinggi, mempunyai adaptasi yang tinggi, lebih tahan terhadap kondisi lingkungan yang kurang baik, kandungan lemak karkas rendah, keempukan daging tidak kalah dengan daging impor [8:15]. Kemampuan reproduksi dari Sapi Bali merupakan yang terbaik diantara kemampuan reproduksi Sapi-Sapi lokal di Indonesia, karena Sapi Bali dapat beranak tiap tahun.

Terdapat beberapa standart penilaian dalam klasifikasi Sapi Bali di Balai Pembibitan Ternak Unggul (BPTU) antara lain [1:3-15]:

## 1. Jantan Sapihan (umur 205 hari)

Tabel 2.2 Standar Penilaian Sapi Bali Jantan Sapihan

No	Uraian	Klasifikasi		
		Baik	Sedang	Buruk
1	Berat Badan (kg)	>100	86-100	<86
2	Warna Kulit	Coklat Kemerahan	Coklat	Coklat Pucat
3	Cacat Tubuh	Tidak Ada	Tidak Ada	Ada
4	Perototan dan Bentuk (cm)			
	- Panjang Badan	>93	87-97	<87
	- Lingkar Dada	>116	108-116	<108
	- Tinggi Gumba	>96	91-96	<91
	- Perototan	Kompak, depan- belakang sesuai	Kompak tapi Tak sesuai	Tak kompak Tak harmonis

Sumber : [1]

## 2. Jantan Umur 1 Tahun

Tabel 2.3 Standar Penilaian Sapi Bali Jantan Umur 1 Tahun

No	Uraian	Klasifikasi		
		Baik	Sedang	Buruk
1	Berat Badan (kg)	>150	125-150	<125
2	Tempramen	Jinak	Terkendali	Liar
3	Warna Kulit	Harmonis	Berubah Sedikit	Tidak Berubah
4	Cacat Tubuh	-	-	-
5	Perototan dan Bentuk (cm)			
	- Panjang Badan	>104	97-104	<97
	- Lingkar Dada	>132	123-132	<123
	- Tinggi Gumba	>106	100-106	<100
	- Perototan	Kompak, depan- belakang sesuai	Kompak tapi Tak sesuai	Tak kompak Tak harmonis

Sumber : [1]

## 3. Jantan Umur 2 Tahun

Tabel 2.4 Standar Penilaian Sapi Bali Jantan Umur 2 Tahun

No	Uraian	Klasifikasi		
		Baik	Sedang	Buruk
1	Berat Badan (kg)	>150	125-150	<125
2	Tempramen	Jinak	Terkendali	Liar
3	Warna Kulit	Harmonis	Berubah Sedikit	Tidak Berubah
4	Cacat Tubuh	-	-	-
5	<i>Scrotum</i>	Besar, kenyal, simetris	Sedang, kenyal, simetris	Kecil, keras, tak simetris
6	Perototan dan Bentuk (cm)			
	- Panjang Badan	>122	115-126	<115
	- Lingkar Dada	>162	156-162	<156
	- Tinggi Gumba	>122	115-122	<115
	- Perototan	Kompak, depan-belakang sesuai	Kompak tapi Tak sesuai	Tak kompak Tak harmonis

Sumber : [1]

## 4. Betina Sapihan (umur 205 hari)

Tabel 2.5 Standar Penilaian Sapi Bali Betina Sapihan

No	Uraian	Klasifikasi		
		Baik	Sedang	Buruk
1	Berat Badan (kg)	>92	78-92	<78
2	Warna Kulit	Coklat Kemerahan	Coklat	Coklat Pucat
3	Cacat Tubuh	Tidak Ada	Tidak Ada	Ada
4	Perototan dan Bentuk (cm)			
	- Panjang Badan	>89	83-89	<83
	- Lingkar Dada	>112	103-112	<103
	- Tinggi Gumba	>93	88-93	<88
	- Perototan	Kompak, depan-belakang sesuai	Kompak tapi Tak sesuai	Tak kompak Tak harmonis

Sumber : [1]

## 5. Betina Umur 1 Tahun

Tabel 2.6 Standar Penilaian Sapi Bali Betina Umur 1 Tahun

No	Uraian	Klasifikasi		
		Baik	Sedang	Buruk
1	Berat Badan (kg)	>136	109-136	<109
2	Tempramen	Jinak	Terkendali	Liar
3	Warna Kulit	Merah Kecoklatan	Coklat	Coklat Pucat
4	Cacat Tubuh	-	-	-
5	Perototan dan Bentuk (cm)			
	- Panjang Badan	>101	95-101	<95
	- Lingkar Dada	>128	116-128	<116
	- Tinggi Gumba	>101	96-101	<96
	- Perototan	Kompak, depan- belakang sesuai	Kompak tapi Tak sesuai	Tak kompak Tak harmonis

Sumber : [1]

## 6. Betina Umur 1.5 Tahun

Tabel 2.7 Standar Penilaian Sapi Bali Betina Umur 1.5 Tahun

No	Uraian	Klasifikasi		
		Baik	Sedang	Buruk
1	Berat Badan (kg)	>190	150-190	<150
2	Tempramen	Jinak	Terkendali	Liar
3	Warna Kulit	Merah Kecoklatan	Coklat	Coklat Pucat
4	Cacat Tubuh	-	-	-
5	Ambing	Besar, harmonis, simetris	Sedang, simetris	Kecil, tak simetris
6	Perototan dan Bentuk (cm)			
	- Panjang Badan	>110	101-110	<101
	- Lingkar Dada	>145	136-145	<136
	- Tinggi Gumba	>112	103-112	<103

- Perototan	Kompak, depan-belakang sesuai	Kompak tapi Tak sesuai	Tak kompak Tak harmonis
-------------	-------------------------------	------------------------	-------------------------

Sumber : [1]

## 2.3 Sistem Pendukung Keputusan (SPK)

Dibutuhkan sebuah sistem yang dapat membantu peternak untuk proses rekomendasi pengambilan keputusan dalam pemilihan bibit unggul Sapi Bali. Sistem ini berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi yaitu sistem bahasa, sistem pengetahuan dan sistem pemrosesan masalah. Pembahasan dalam sub bab ini meliputi karakteristik dan kemampuan SPK serta komponen yang terdapat pada SPK.

### 2.3.1 Karakteristik dan Kemampuan SPK

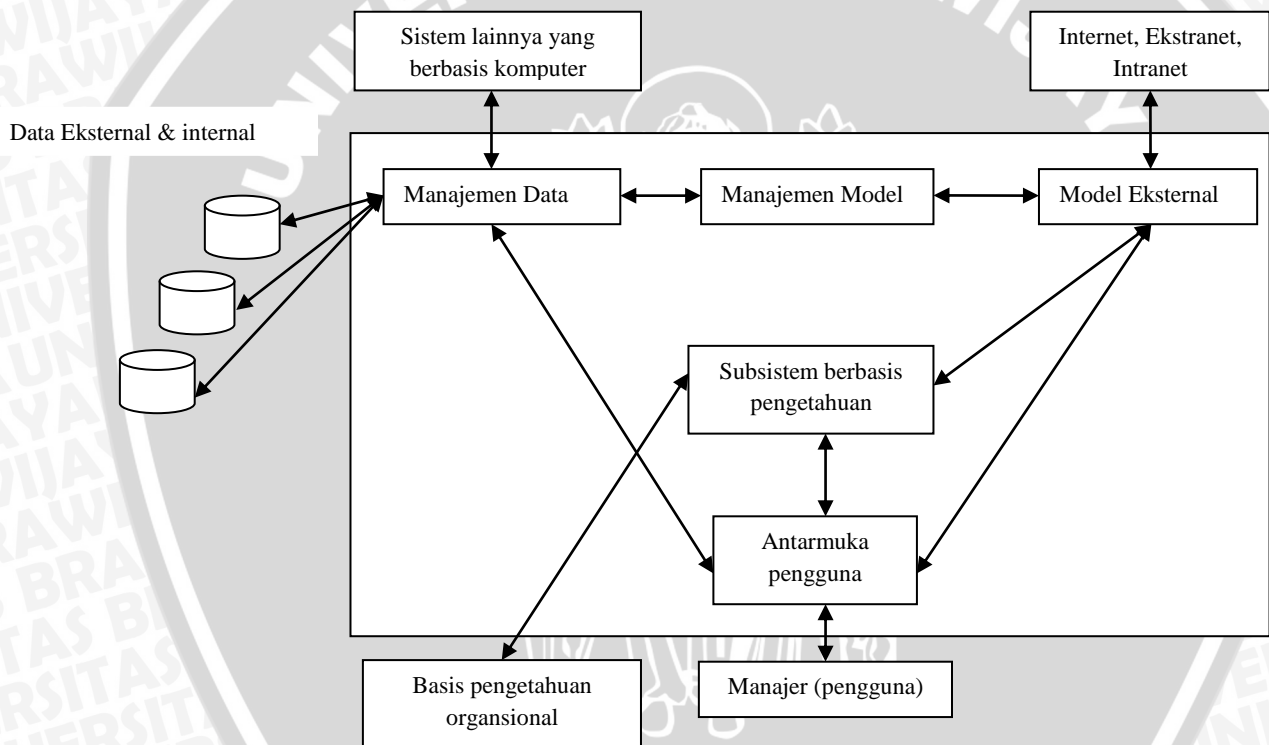
Karakteristik dan kemampuan yang dimiliki oleh SPK adalah sebagai berikut [29:140-143]:

1. SPK membantu mengambil keputusan, baik dalam kondisi semi-terstruktur ataupun tidak terstruktur.
2. SPK dapat mendukung serta digunakan oleh berbagai lapisan, mulai dari lapisan manager hingga karyawan.
3. SPK dapat digunakan untuk semua individu dan kelompok.
4. SPK menyediakan keputusan yang bersifat keputusan independen ataupun sekuensial.
5. Proses pengambilan keputusan dilakukan pada fase : *intelegensi, desain, pilihan, dan implementasi*
6. SPK dapat menghasilkan berbagai macam gaya keputusan.
7. SPK harus fleksibel dan mudah diadaptasikan.
8. Pengguna harus merasa nyaman dalam menggunakan SPK.
9. SPK mampu meningkatkan keefektifan dalam pengambilan keputusan (*accuracy, timeliness, quality*).
10. SPK hanya memberikan saran dan bukan keputusan mutlak dalam sebuah pengambilan keputusan.
11. Pengguna SPK bisa mengembangkan dan memodifikasi sendiri sistem sederhana sesuai dengan keinginannya.

12. SPK biasa digunakan sebagai model untuk menganalisa keputusan yang diambil.
13. SPK menyediakan akses untuk berbagai sumber data, format, dan tipe.
14. SPK dapat diintegrasikan dengan SPK yang lain dan atau aplikasi lain, serta dapat didistribusikan secara internal dan eksternal menggunakan networking dan teknologi Web.

### 2.3.2 Komponen SPK

Secara umum perancangan sistem pendukung keputusan mencakup 3 komponen utama manajemen data, manajemen model dan manajemen antarmuka. Arsitektur sistem pendukung keputusan dapat dilihat pada Gambar 2.2



Gambar 2.2 Arsitektur Pemodelan Sistem Pendukung Keputusan  
Sumber: [20:144]

SPK dapat terdiri dari empat subsistem utama yang menentukan kapabilitas teknis sistem pendukung keputusan yaitu [20:143-145]:

1. Subsistem Manajemen Berbasis Pengetahuan

Subsistem ini dapat mendukung semua subsistem lain atau bertindak sebagai suatu komponen independen.

## 2. Subsistem Manajemen Data

Subsistem manajemen data memasukan suatu database yang berisi data yang relevan untuk situasi dan dikelola oleh perangkat lunak yang disebut dengan sistem manajemen database (DBMS). Subsistem manajemen data dapat diinterkoneksi dengan data *warehouse* perusahaan untuk pengambilan keputusan.

## 3. Subsistem Manajemen Model

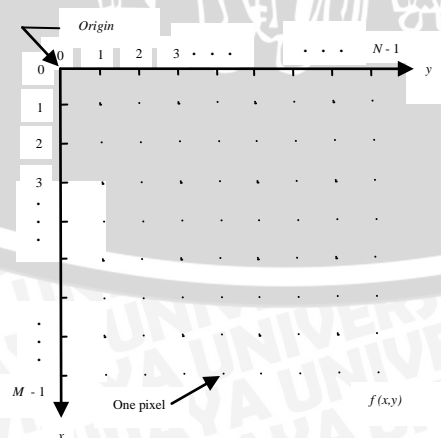
Sistem pendukung keputusan dapat mengintegrasikan data dengan model-model keputusan. Model adalah suatu tiruan dari alam nyata, tetapi model tidak mampu mencerminkan seluruh variabel alam nyata sehingga keputusan yang diambil tidak sesuai dengan kebutuhan.

## 4. Subsistem Antarmuka Pengguna

Pengguna berkomunikasi dengan dan memerintahkan sistem pendukung keputusan melalui subsistem ini. Pengguna adalah bagian yang dipertimbangkan dari sistem.

### 2.4 Citra Digital

Citra digital dapat didefinisikan sebagai fungsi dua variabel  $f(x,y)$ , dimana  $x$  dan  $y$  adalah koordinat opasial dan nilai  $f(x,y)$  adalah intensitas citra pada koordinat tersebut [7:1]. Citra memiliki karakteristik yang sangat penting yaitu warna. Warna yang terdapat pada citra digital merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau dan biru (*Red, Green, Blue* - RGB). Sistem koordinat pada sebuah citra digital dapat dilihat pada Gambar 2.3.



Gambar 2.3 Koordinat Citra Digital  
Sumber : [7:55]

### 2.4.1 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. Angka 1 mewakili warna hitam dan angka 0 mewakili warna putih [7:522]. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap *pixel* dari citra biner.



Gambar 2.4 Citra biner  
Sumber : [Observasi]

### 2.4.2 Citra Grayscale

Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixelnya, dengan kata lain nilai bagian red=green= blue. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Tingkatan keabuan di sini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih. Citra *grayscale* berikut memiliki ke dalaman warna 8 bit (256 kombinasi warna keabuan).



Gambar 2.5 Citra grayscale  
Sumber : [Observasi]



### 2.4.3 Citra Warna

Citra berwarna, yaitu citra digital yang setiap *pixel*-nya mengandung informasi warna. Informasi warna ini biasanya terbentuk dari tiga komponen warna. Tiga komponen warna yang digunakan adalah *Red-Green-Blue* (RGB), tetapi dalam konteks lain sering juga digunakan komponen warna yang lain seperti *Cyan-Magenta-Yellow-Black* (CMYK) atau YcbCr [7:289-295].



Gambar 2.6 Citra warna  
Sumber : [Observasi]

### 2.5 Klasifikasi

Klasifikasi adalah suatu proses membuat model atau fungsi untuk mengelompokkan sejumlah data ke dalam kelas tertentu berdasarkan kelakuan dan atribut yang terapat dalam data [14:5]. Tujuan dari proses ini adalah untuk dapat memperkirakan kelas dari suatu objek yang kelasnya tidak diketahui.

Proses klasifikasi mempunyai dua data yang berbeda yaitu *data training* dan *data testing*. *Data training* digunakan sebagai acuan untuk membuat aturan pada klasifikasi data. *Data testing* digunakan sebagai uji coba pada data yang belum mempunyai kelas berdasarkan aturan yang dibentuk dari *data training*.

Proses klasifikasi biasanya dibagi menjadi dua fase, yaitu [13:286-289]:

#### 1. *Learning*

Pada fase ini sebagian data yang telah diketahui kelasnya akan dibuat sebuah model yang digunakan untuk perkiraan.

#### 2. *Test*

Pada fase model yang sudah terbentuk diuji dengan sebagian data lainnya untuk mengetahui akurasi dari model tersebut. Jika akurasinya mencukupi, model ini dapat digunakan untuk prediksi kelas data yang belum diketahui.

Proses klasifikasi terdapat beberapa metode yang digunakan, antara lain *Decision Tree*, *Naive Bayes*, *Neural Network*, *Support Vector Machine (SVM)* serta *KNN*. *Support Vector Machine (SVM)* adalah salah satu metode klasifikasi yang berbasis *machine learning* yang memiliki akurasi yang baik dalam proses klasifikasi [3:203].

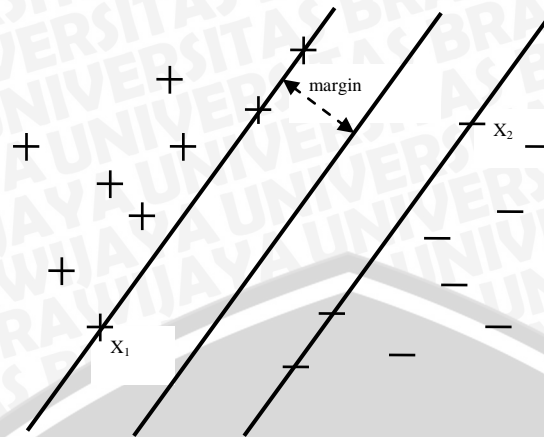
## 2.6 *Support Vector Machine (SVM)*

SVM merupakan salah satu metode yang dapat digunakan dalam proses klasifikasi dan termasuk ke dalam kelas *supervised learning*. Metode ini merupakan metode yang baru dan memiliki tingkat performansi yang baik dibandingkan dengan metode klasifikasi lainnya seperti *Naive Bayes*, *C.45* dan *KNN* [3:203]. Prinsip dasar dari metode ini adalah *binary classifier*, yang berarti hanya dapat mengklasifikasikan suatu kasus ke dalam dua kelas yang berbeda (+1 dan -1)[5:3]. Tujuan utama dari metode ini adalah mencari *hyperplane* (garis pemisah) terbaik antar dua kelas [2:3-4].

Pada permasalahan *nonlinear SVM* menggunakan sebuah pemetaan *nonlinear* yang disebut fungsi kernel untuk mengubah data training asli ke dalam ruang fitur (*feature space*) berdimensi tinggi. Penelitian ini akan dilakukan secara *nonlinear* proses klasifikasi. Fungsi pemisah yang akan dipisahkan oleh bidang pembatas berupa suatu fungsi *nonlinear*.

### 2.6.1 *SVM Linear*

Data pada SVM linear merupakan data yang dapat dipisahkan secara *linear*. Misalkan  $x_i \in \{x_1, \dots, x_n\}$  adalah dataset dan  $y_i \in \{+1, -1\}$  adalah label kelas data  $x_i$ . Pada Gambar 2.7 dapat dilihat bahwa terdapat bidang pemisah (*hyperplane*) yang memisahkan semua data set sesuai dengan kelasnya. *Hyperplane* yang terbaik memiliki margin yang paling besar dalam memisahkan data set [5:1].



Gambar 2.7 Hyperplane yang memisahkan dua kelas  
 Sumber : [5:3]

Data set dipisahkan oleh sepasang bidang pembatas yang sejajar menjadi dua kelas. Persamaan untuk bidang pembatas tersebut ditunjukkan pada persamaan (2-1):

$$\begin{aligned} x_i \cdot w + b &\geq +1 \text{ untuk } y_i = +1 \\ x_i \cdot w + b &\leq -1 \text{ untuk } y_i = -1, \dots \dots \dots (2-1) \end{aligned}$$

dimana:

- $w$  = vektor bobot
- $x_i$  = data ke- $i$ ,  $i = 1, 2, \dots, n$
- $b$  = bilangan skalar yang menyatakan nilai bias
- $y_i$  = target/kelas (sampel positif (+1) dan sampel negatif(-1))

Nilai *margin* antara dua kelas dapat dihitung dengan mencari jarak antara kedua bidang pemisah (*hyperplane*) dari dua kelas. Persamaan yang digunakan untuk mencari nilai *margin* antara dua kelas sebagai berikut:

$$\begin{aligned} (x_i \cdot w + b = +1) - (x_i \cdot w + b = -1) \\ w(x_1 - x_2) &= 2 \\ \frac{w}{\|w\|} (x_1 - x_2) &= \frac{2}{\|w\|} \dots \dots \dots (2-2) \end{aligned}$$

dimana:

- $\|w\| = \sqrt{W^T \cdot W} = \sqrt{w_1^2 + w_2^2 + \dots + w_n^2}$
- $w$  = vektor bobot
- $x_i$  = data ke- $i$ ,  $i = 1, 2, \dots, n$
- $b$  = bilangan skalar yang menyatakan nilai bias

Nilai margin dimaksimalkan dengan tetap memenuhi persamaan (2-1) dengan mengalikan  $b$  dan  $w$  dengan sebuah konstanta. Nilai *margin* terbesar dapat dirumuskan menjadi permasalahan optimasi konstrain ke dalam persamaan (2-3) dengan memaksimalkan  $\frac{1}{\|w\|}$  atau meminimumkan  $\|w\|^2$  dengan memperhatikan bidang pembatas pada persamaan (2-1) .

$$\min \frac{1}{2} \|w\|^2 \dots\dots\dots(2-3)$$

dimana:

$\|w\|$  = *Euclidean Norm* dari  $W$

*Lagrange Multipliers* merupakan salah satu teknik komputasi yang dapat digunakan untuk menyelesaikan permasalahan optimasi konstrain. Persamaan (2-3) dapat diubah menjadi persamaan (2-4) *primal problem* ( $L_p$ ) melalui teknik komputasi *Lagrange Multipliers*.

$$\min_{w,b} L_p(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha_i y_i (x_i \cdot w_i + b) + \sum_{i=1}^l \alpha_i \dots\dots\dots(2-4)$$

dimana:

$w$  = vektor bobot

$l$  = jumlah banyaknya data

$\alpha_i$  = *Lagrange Multipliers* ;  $\alpha_i \geq 0$

$b$  = bilangan skalar yang menyatakan nilai bias

$x_i$  = data ke- $i$ ,  $i = 1, 2, \dots, n$

Proses meminimalkan persamaan (2-4) akan didapatkan dua kondisi optimasi pada persamaan (2-5) dan persamaan (2-6).

$$\text{Kondisi 1: } \frac{\delta}{\delta w} L_p(w, b, \alpha) = 0 \dots\dots\dots(2-5)$$

$$\text{Kondisi 2 : } \frac{\delta}{\delta b} L_p(w, b, \alpha) = 0 \dots\dots\dots(2-6)$$

Menerapkan kondisi 1 dan kondisi 2 pada fungsi *Lagrange Multipliers* (2-4) maka didapatkan persamaan (2-7) dan persamaan (2-8):

$$w = \sum_{i=1}^n \alpha_i y_i x_i , \dots\dots\dots(2-7)$$

$$\sum_{i=1}^n \alpha_i y_i = 0, \dots\dots\dots(2-8)$$

dimana:



$w$  = vektor bobot

$y_i$  = target/kelas (sampel positif (+1) dan sampel negatif(-1))

Nilai vektor  $w$  sering kali bernilai besar sampai tak terhingga sedangkan nilai  $\alpha_i$  terhingga. Persamaan (2-4) diubah ke dalam *dual problem* ( $L_D$ ) dengan mensubstitusikan persamaan (2-7) dan persamaan (2-8) ke persamaan (2-4) sehingga diperoleh persamaan (2-9).

$$L_d(\alpha) = \frac{1}{2} W^T W - \sum_{i=1}^l \alpha_i y_i w^T x_i - b \sum_{i=1}^l \alpha_i y_i + \sum_{i=1}^l \alpha_i$$

$$L_d(\alpha) = \frac{1}{2} \sum_{i=1}^l \alpha_i y_i W^T x_i - \sum_{i=1}^l \alpha_i y_i w^T x_i - b \sum_{i=1}^l \alpha_i y_i + \sum_{i=1}^l \alpha_i$$

$$L_d(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \alpha_i \alpha_j y_i y_j x_i^T x_j \dots \dots \dots (2-9)$$

dimana:

$w$  = vektor bobot

$l$  = jumlah banyaknya data

$\alpha_i$  = *Lagrange Multipliers* ;  $\alpha_i \geq 0$

$b$  = bilangan skalar yang menyatakan nilai bias

$x_i$  = data ke- $i$ ,  $i = 1, 2, \dots, n$

Hasil dari persamaan (2-9) ialah nilai  $\alpha_i$  yang nantinya digunakan untuk menemukan nilai  $w$  dan  $b$  yang optimal. Persamaan (2-10) dan (2-11) merupakan persamaan untuk mendapatkan nilai  $w$  dan  $b$  yang optimal.

$$w = \sum_{i=1}^l \alpha_i x_i y_i \dots \dots \dots (2-10)$$

$$b = \frac{1}{N_{sv}} \left( \sum_{s=1}^{N_{sv}} \left( \frac{1}{y_s} - x_s^T w \right) \right) \dots \dots \dots (2-11)$$

dimana:

$w$  = vektor bobot

$l$  = jumlah banyaknya data

$\alpha_i$  = *Lagrange Multipliers* ;  $\alpha_i \geq 0$

$b$  = bilangan skalar yang menyatakan nilai bias

$N_{sv}$  = Jumlah support vektor

Data yang memiliki nilai  $\alpha_i > 0$  adalah *support vector*. Fungsi keputusan yang dihasilkan hanya dipengaruhi oleh *support vector*. Setelah solusi permasalahan



*quadratic programming* ditemukan (nilai  $\alpha_i$ ), maka kelas dari data pengujian dapat ditentukan berdasarkan nilai dari fungsi keputusan.

Fungsi keputusan yang digunakan untuk memisahkan kelas adalah menggunakan fungsi *linear*, dimana fungsi tersebut didefinisikan sebagai berikut:

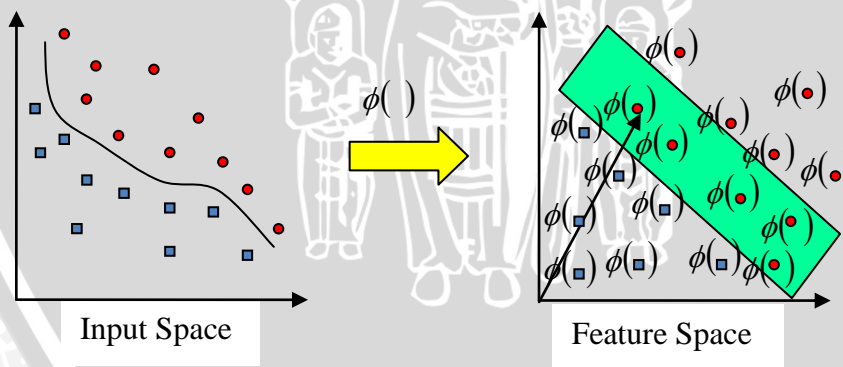
$$f(x_d) = \text{sign}(\sum_{i=1}^n \alpha_i y_i x_i \cdot x_d + b), \dots\dots\dots (2-12)$$

dimana:

- $n$  = jumlah *support vector*
- $x_d$  = data yang akan diklasifikasikan
- $y_i$  = kelas (sampel positif (+1) dan sampel negatif(-1)) dari *support vector*
- $b$  = bilangan skalar yang menyatakan nilai *bias*

**2.6.2 SVM NonLinear**

Konsep dasar dari SVM *nonlinear* adalah mentransformasikan data ke dalam dimensi ruang fitur (*feature space*) sehingga dapat dipisahkan secara *linear* pada *feature space*. Data dipetakan menggunakan fungsi pemetaan (transformasi)  $x_k \rightarrow \phi(x_k)$  ke dalam *feature space* sehingga terdapat bidang pemisah yang dapat memisahkan data sesuai dengan kelasnya. Gambar 2.8 menunjukkan proses transformasi data ke dalam *feature space*.



Gambar 2.8 Transformasi data ke *feature space*  
 Sumber:[19:8]

Penggunaan fungsi transformasi  $x_k \rightarrow \phi(x_k)$  akan menghasilkan fungsi pembelajaran dan  $w$  ditunjukkan pada persamaan (2-13) dan (2-14).

$$w = \sum_{i=1}^n \alpha_i y_i \phi(x_i), \dots\dots\dots (2-13)$$

dimana:

- $w$  = vektor bobot



$y_i$  = target/kelas (sampel positif (+1) dan sampel negatif(-1))

$\phi(x_i)$  = Transformasi data ke-  $i$

$n$  = jumlah *support vector*

$$f(x_d) = \sum_{i=1}^n \alpha_i y_i \phi(x_i) \cdot \phi(x_d) + b, \dots\dots\dots (2-14)$$

dimana:

$y_i$  = target/kelas (sampel positif (+1) dan sampel negatif(-1))

$\phi(x_i)$  = Transformasi data ke-  $i$

$n$  = jumlah *support vector*

$b$  = bilangan skalar yang menyatakan nilai bias

Proses transformasi  $\Phi$  memiliki ukuran yang sangat besar dan memiliki jumlah *feature* yang tidak terhingga, sehingga sangat sulit untuk dipahami. Permasalahan tersebut dapat diselesaikan menggunakan fungsi kernel. Pada persamaan (2-12) terdapat *dot product* ( $x_i \cdot x_d$ ), dengan adanya penggunaan fungsi kernel  $K$  maka didapatkan fungsi keputusan yang ditunjukkan pada persamaan (2-15).

$$f(x_d) = \text{sign}(\sum_{i=1}^n \alpha_i y_i K(x_i, x_d) + b), \dots\dots\dots (2-15)$$

dimana:

$K(x_i, x_d)$  = fungsi kernel,  $K(x_i, x_d) = \phi(x_i) \cdot \phi(x_d)$ .

$y_i$  = target/kelas (sampel positif (+1) dan sampel negatif(-1))

$n$  = jumlah *support vector*

$b$  = bilangan skalar yang menyatakan nilai bias

Terdapat beberapa fungsi kernel yang sering digunakan dalam penyelesaian permasalahan *nonlinear* yang ditunjukkan dalam persamaan [22:28]:

1. *Linear* atau *dot product* yang didefinisikan dengan persamaan (2-16)

$$K(x_i, x_j) = (x_i^T x_j) \dots\dots\dots (2-16)$$

2. *Gaussian RBF* yang didefinisikan dengan persamaan (2-17)

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad \gamma > 0 \dots\dots\dots (2-17)$$

3. *Polynomial* dengan derajat  $d$  didefinisikan dengan persamaan (2-18)

$$K(x_i, x_j) = (\gamma(x_i^T x_j) + r)^d \dots\dots\dots (2-18)$$

4. *Sigmoid* yang didefinisikan dengan persamaan (2-19)

$$K(x_i, x_j) = \frac{1}{1 + \exp(-(x_i^T x_j + r))} \dots \dots \dots (2-19)$$

Pada penelitian ini menggunakan fungsi kernel *Gaussian RBF*, dikarenakan fungsi kernel tersebut memiliki tingkat performansi yang sama dengan kernel *linear* pada parameter tertentu dan rentang nilainya kecil [12:4].

## 2.7 Multi-Class SVM

SVM pertama kali diperkenalkan oleh Vapnik, SVM hanya dapat mengklasifikasikan data ke dalam dua kelas (*binary classifier*). SVM dapat diterapkan pada pengklasifikasian data yang bersifat *multi-class*. Pada implementasinya ada dua pendekatan yang umumnya dilakukan. Pendekatan pertama yaitu menggabungkan beberapa SVM biner dan pendekatan yang kedua dengan menggabungkan semua data yang terdiri dari beberapa kelas ke dalam permasalahan optimasi [19:10].

Pada penggunaannya, pendekatan yang kedua jauh lebih rumit dalam hal permasalahan optimasi yang harus diselesaikan. Pendekatan yang pertama pada umumnya menggunakan metode untuk mengimplementasikan *multi-class SVM* antara lain, metode *one-against-all*, *one-against-one* dan *Directed Acyclic Graph Support Vector Machine (DAGSVM)*. Penelitian ini akan menerapkan metode yang digunakan untuk permasalahan *multi-class* yaitu metode *One-against-all*.

### 2.7.1 Metode *one-against-all*

Penelitian ini akan menggunakan metode *one-against-all* untuk menangani permasalahan *multi-class*. Metode ini dibangun berdasarkan sejumlah  $k$  SVM biner ( $k$  adalah jumlah kelas) [11:415-416]. SVM ke- $i$  dilatih dengan seluruh data sampel yang terdapat pada ke- $i$  dengan memiliki label kelas positif (+1) dan label kelas negatif (-1). Misalkan terdapat permasalahan klasifikasi dengan 3 buah kelas, untuk pelatihan pada kelas-kelas tersebut akan menggunakan 3 buah SVM biner seperti pada Tabel 2.8.

$$\min_{w^i, b^i, \xi^i} \frac{1}{2} (w^i)^T w^i + C \sum_{j=1}^1 \xi_j^i,$$

$$(w^i)^T \phi(X_j) + b^i \geq 1 \xi_j^i, \text{ jika } y_j = i,$$



$$(w^i)^T \phi(X_j) + b^i \leq -1\xi_j^i, \text{ jika } j \neq i$$

$$\xi_j^i \geq 0, j = 1, \dots, l \dots \dots \dots (2-20)$$

dimana:

k = jumlah kelas

$x_i^T x_j$  = dot product dari data input

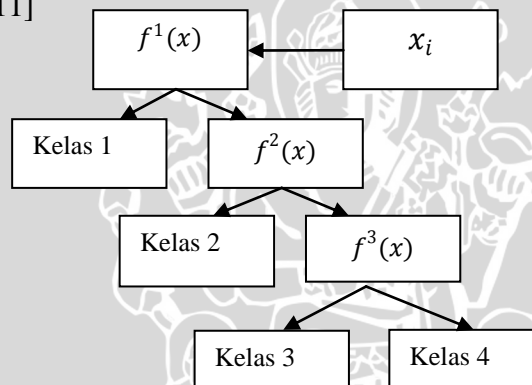
$\xi$  = nilai penalti pada data yang tidak memenuhi pembatas

C = batasan error bernilai positif

Tabel 2.8 Metode *One-against-all* dengan 4 Kelas

$y_i = 1$	$y_i = -1$	Hipotesis
Kelas 1	Bukan kelas 1	$f^1(x) = (w^1)x + b^1$
Kelas 2	Bukan kelas 2	$f^2(x) = (w^2)x + b^2$
Kelas 3	Kelas 4	$f^3(x) = (w^3)x + b^3$

Sumber: [19:11]



Gambar 2.9 Contoh klasifikasi *One Against All* untuk 4 kelas  
Sumber : [19:11]

**2.7.2 Metode *one-against-one***

Metode ini dibangun berdasarkan  $k(k - 1)/2$  buah model klasifikasi SVM biner [11:415-416]. Pada metode ini dilatih dari dua kelas yaitu kelas ke- $i$  dan kelas ke- $j$ , dilakukan pencarian solusi untuk persoalan optimasi konstrain sebagai berikut:

$$\min_{w^{i,j}, b^{i,j}, \xi^{i,j}} \frac{1}{2} (w^{i,j})^T w^{i,j} + C \sum_{j=1}^1 \xi_t^{i,j},$$

$$(w^{i,j})^T \phi(X_t) + b^{i,j} \geq 1\xi_t^{i,j}, \text{ jika } y_t = i,$$

$$(w^{i,j})^T \phi(X_t) + b^{i,j} \leq -1\xi_t^{i,j}, \text{ jika } y_t \neq i$$

$$\xi_t^{i,j} \geq 0 \dots \dots \dots (2-21)$$

dimana:

$k$  = jumlah kelas

$x_i^T x_j$  = dot product dari data input

$\xi$  = nilai penalti pada data yang tidak memenuhi pembatas

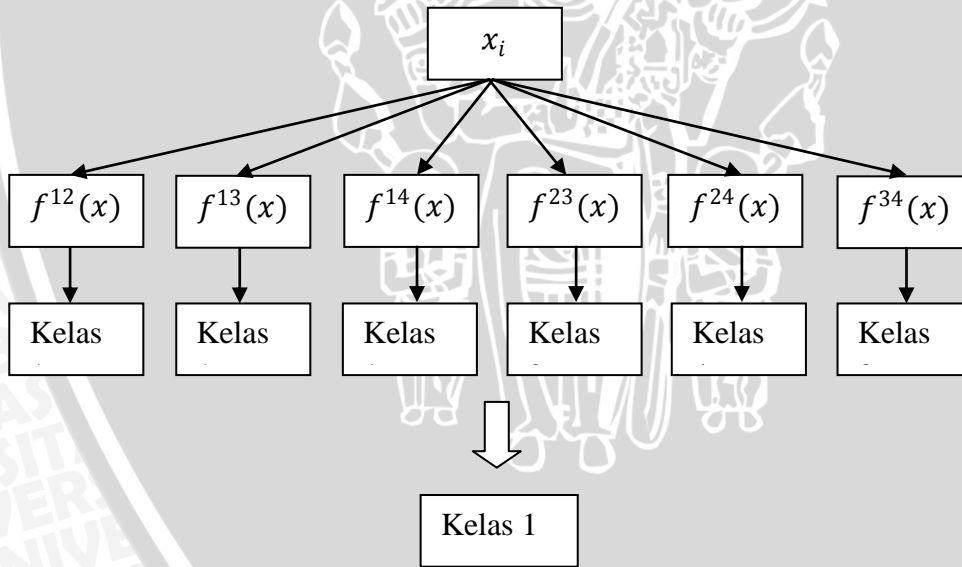
$C$  = batasan error bernilai positif

Terdapat beberapa metode untuk melakukan pengujian setelah keseluruhan  $k(k - 1)/2$  model klasifikasi selesai dibangun. Salah satu dari metode tersebut adalah metode *voting*.

Tabel 2.9 Metode *One-Against-One* dengan 6 SVM biner

$y_i = 1$	$y_i = -1$	Hipotesis
Kelas 1	Bukan kelas 2	$f^{12}(x) = (w^{12})x + b^{12}$
Kelas 1	Bukan kelas 3	$f^{13}(x) = (w^{13})x + b^{13}$
Kelas 1	Bukan kelas 4	$f^{14}(x) = (w^{14})x + b^{14}$
Kelas 2	Bukan kelas 3	$f^{23}(x) = (w^{23})x + b^{23}$
Kelas 2	Bukan kelas 4	$f^{24}(x) = (w^{24})x + b^{24}$
Kelas 3	Bukan kelas 4	$f^{34}(x) = (w^{34})x + b^{34}$

Sumber: [19:12]



Gambar 2.10 Contoh klasifikasi dengan metode *One-against-one*

Sumber: [19:13]

Berdasarkan Gambar 2.10, data  $x_i$  dapat diklasifikasikan sebagai kelas 1 karena kelas 1 mempunyai suara terbanyak. Jika data  $x_i$  dimasukkan ke dalam fungsi hasil pelatihan pada persamaan berikut :

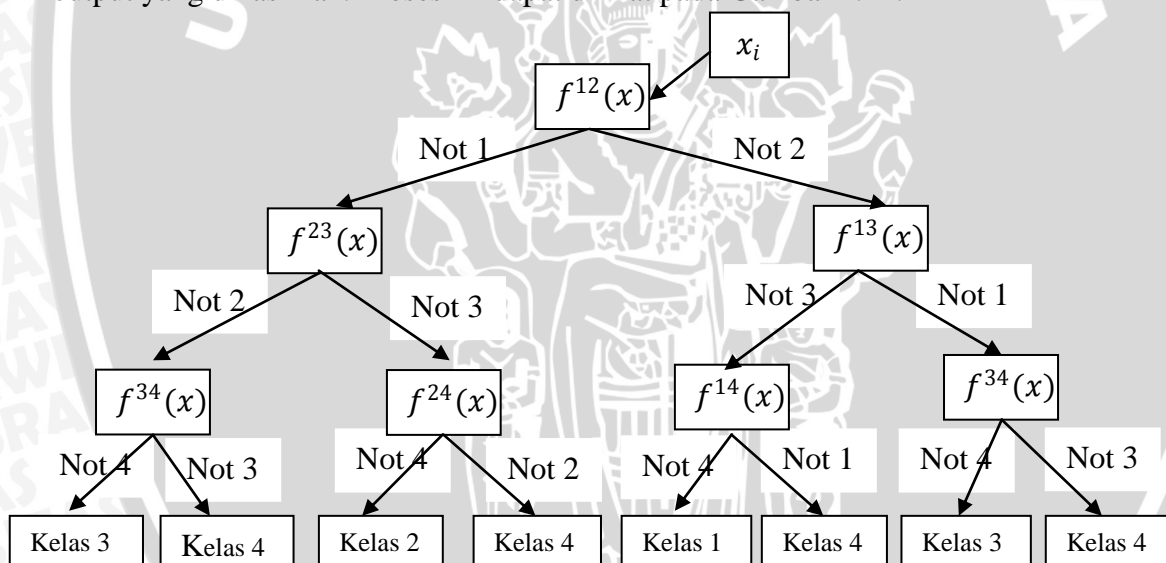
$$f(x) = (w^{ij})^T \phi(X) + b \dots \dots \dots (2-22)$$



Hasilnya menyatakan bahwa data  $x_i$  adalah kelas 1, maka suara untuk kelas 1 ditambah satu. Kelas dari data  $x_i$  akan ditentukan dari jumlah suara terbanyak. Jika terdapat dua buah kelas yang memiliki jumlah suara yang sama, maka kelas yang indeksnya lebih kecil dinyatakan sebagai kelas dari data  $x_i$ . Jadi pada metode ini terdapat  $k(k-1)/2$  buah permasalahan *quadratic programming* yang masing-masing memiliki  $2n/k$  variabel ( $n$  adalah jumlah data pelatihan).

### 2.7.3 Metode *Directed Acyclic Graph Support Vector Machine (DAGSVM)*

Metode DAGSVM dibangun berdasarkan  $k(k-1)/2$  buah model klasifikasi SVM biner [11:415-416]. Proses pelatihan pada metode ini sama seperti metode *one-against-one*, akan tetapi pada saat pengujian menggunakan *binary directed acyclic graph*. Pada proses memprediksi kelas data pengujian, pengujian dilakukan mulai dari simpul akar dan kemudian bergerak ke kiri atau ke kanan tergantung dari nilai output yang dihasilkan. Proses ini dapat dilihat pada Gambar 2.11.



Gambar 2.11 Contoh klasifikasi dengan metode DAGSVM  
Sumber: [19:14]

Tabel 2.10 Metode DAGSVM dengan 6 SVM Biner

$y_i = 1$	$y_i = -1$	Hipotesis
Bukan kelas 2	Bukan kelas 1	$f^{12}(x) = (w^{12})x + b^{12}$
Bukan kelas 3	Bukan kelas 1	$f^{13}(x) = (w^{13})x + b^{13}$
Bukan kelas 4	Bukan kelas 1	$f^{14}(x) = (w^{14})x + b^{14}$
Bukan kelas 3	Bukan kelas 2	$f^{23}(x) = (w^{23})x + b^{23}$
Bukan kelas 4	Bukan kelas 2	$f^{24}(x) = (w^{24})x + b^{24}$
Bukan kelas 4	Bukan kelas 3	$f^{34}(x) = (w^{34})x + b^{34}$

Sumber : [19:13]

## 2.8 Sequential Minimal Optimization (SMO)

SMO merupakan algoritma yang digunakan dalam proses pelatihan SVM yang dapat memberikan solusi pada permasalahan kuadratik (*Quadratic Problem*). Pada permasalahan kuadratik, SMO akan memilih untuk menyelesaikan kemungkinan optimasi terkecil untuk setiap tahapnya. Setiap tahap, algoritma SMO memilih dua *Lagrange Multipliers*  $\alpha_i$  untuk dioptimisasi secara bersama-sama, mencari nilai paling optimal untuk *lagrange multipliers*, dan memperbarui SVM dengan nilai optimal yang baru.

SMO memiliki tiga komponen utama antara lain metode penyelesaian secara analitik untuk menyelesaikan dua *Lagrange Multipliers*, pemilihan secara heuristik untuk memilih *Lagrange* yang akan dioptimalkan, dan metode untuk menghitung nilai *bias* ( $b$ ) [17:45].

Tahapan komponen pertama pada SMO merupakan penyelesaian untuk dua *Lagrange Multipliers* adalah sebagai berikut [17:46]:

1. Metode penyelesaian pengali *Lagrange* dimulai dengan menghitung pengali *lagrange* kedua ( $\alpha_2$ ) dan menghitung segmen garis diagonalnya.

$$L = \begin{cases} \max(0, \alpha_2^{lama} - \alpha_1^{lama}); & \text{Jika } y_1 \neq y_2 \\ \max(0, \alpha_2^{lama} + \alpha_1^{lama} - C); & \text{Jika } y_1 = y_2 \end{cases} \dots\dots\dots(2-23)$$

$$H = \begin{cases} \min(C, C + \alpha_2^{lama} - \alpha_1^{lama}); & \text{Jika } y_1 \neq y_2 \\ \min(C, \alpha_2^{lama} + \alpha_1^{lama}); & \text{Jika } y_1 = y_2 \end{cases} \dots\dots\dots(2-24)$$

dimana:

$\alpha_i$  = *Lagrange Multipliers*

$y_i$  = target/kelas (sampel positif (+1) dan sampel negatif(-1))

$C$  = batasan *error* bernilai positif

Sehingga turunan kedua dari fungsi objektif sepanjang garis diagonal dirumuskan dengan persamaan (2-25):

$$\eta = 2K(\vec{x}_1, \vec{x}_2) - K(\vec{x}_1, \vec{x}_1) - K(\vec{x}_2, \vec{x}_2) \dots\dots\dots(2-25)$$

dimana:

$K(\vec{x}_1, \vec{x}_1)$  = fungsi kernel

- Menghitung lokasi batas maksimal dari fungsi objektif selama hanya terdapat dua pengali *Lagrange*. Pada keadaan normal terdapat nilai maksimal sepanjang fungsinya dari batas persamaan linier, sehingga nilai  $\eta < 0$ . SMO akan menghitung batas maksimal dengan persamaan (2-26):

$$\alpha_2^{baru} = \alpha_2^{lama} - \frac{y_2(E_1 - E_2)}{\eta} \dots \dots \dots (2-26)$$

dimana:

$$f^{lama}(\vec{x}_i) = \sum_{i=1}^n \alpha_i y_i K(x, x_i) - b$$

$$E_i = f^{lama}(\vec{x}_i) - y_i \dots \dots \dots (2-27)$$

$E_i$  = error pada data training ke-i

- Mencari batasan maksimum dengan memotong pada segmen garis selain batas maksimum dengan persamaan (2-28):

$$\alpha_2^{baru, potongan} = \begin{cases} H, & \text{if } \alpha_2^{baru} > H; \\ \alpha_2^{baru}, & \text{if } L < \alpha_2^{baru} < H; \dots \dots \dots (2-28) \\ L, & \text{if } \alpha_2^{baru} < L; \end{cases}$$

Hitung nilai  $\alpha_1^{baru}$  dengan persamaan 2-28:

$$\alpha_1^{baru} = \alpha_1^{lama} + y_1 * y_2 (\alpha_2^{lama} - \alpha_2^{baru, potongan}) \dots \dots \dots (2-29)$$

dimana:

$\alpha_i$  = *Lagrange Multipliers*

$y_i$  = kelas (sampel positif (+1) dan sampel negatif(-1))

Tahapan komponen kedua yaitu pemilihan secara heuristik untuk memilih *Lagrange* yang akan dioptimalkan. Pada tahap ini akan dibagi menjadi dua yaitu pengali *Lagrange* pertama dan pengali *Lagrange* kedua. Pemilihan secara heuristik adalah pemilihan heuristik dilakukan dengan *outer loop* dari algoritma SMO. Proses ini akan dilakukan pada data *training*, *outer loop* tidak selalu melakukan iterasi ke seluruh data *training*. *Outer loop* akan melakukan iterasi pada data *training* yang memenuhi kondisi  $\alpha_i > 0$  dan  $\alpha_i < C$ . Setiap data *training* dicek dengan kondisi *Karush-Kuhn-Tucker* (KKT), data yang melanggar kondisi tersebut akan dioptimasi dan diupdate. *Outer loop* diulang melewati seluruh data yang tidak melwati batas sampai data yang keluar batas memenuhi kondisi KKT dengan nilai  $\epsilon$  (epsilon) yang telah



ditentukan. Biasanya nilai dari epsilon berkisar antara  $10^{-2}$  sampai  $10^{-3}$ . *Outer loop* akan melewati seluruh data *training subset* yang keluar batas sampai data *training* memenuhi kondisi KKT dan algoritma SMO akan berhenti. Penentuan pengali *Lagrange* kedua ini dilakukan secara acak dan akan ikut dimaksimalkan selama proses optimasi dengan *lagrange* pertama. Evaluasi fungsi kernel K membutuhkan waktu, sehingga SMO akan melakukan langkah selanjutnya dengan nilai *absolut* dari selisih nilai  $|E_1 - E_2|$ .

Tahap ketiga komponen SMO merupakan proses untuk mencari nilai bias (b) SVM [17::48]. Proses penyelesaian pengali *Lagrange*  $\alpha_i$  tidak menghitung bias (b) dari SVM, sehingga bias dihitung secara terpisah. Pada setiap langkah, bias akan dihitung ulang, sehingga kondisi KKT dipenuhi oleh kedua data yang dioptimasi. Nilai bias  $b_1$  valid jika  $\alpha_1$  yang baru tidak pada batas karena akan menyebabkan output SVM  $y_1$  jika output  $x_1$  yang diformulasikan pada persamaan (2-29):

$$b_1 = E_1 + y_1(\alpha_1^{baru} - \alpha_1^{lama})K(\vec{x}_1, \vec{x}_1) + y_2(\alpha_2^{baru, potongan} - \alpha_2^{lama})K(\vec{x}_1, \vec{x}_2) + b^{lama} \dots\dots\dots(2-30)$$

dimana:

- $\alpha_i$  = *Lagrange Multipliers*
- $y_i$  = kelas (sampel positif (+1) dan sampel negatif(-1))
- $K(\vec{x}_1, \vec{x}_1)$  = fungsi kernel
- $b$  = bias
- $E_i$  = error pada data training ke-i

Nilai  $b_2$  valid jika  $\alpha_2$  yang baru tidak berada pada batas, karena akan menyebabkan output SVM menjadi  $y_2$  jika input  $x_2$  yang diformulasikan pada persamaan (2-31):

$$b_2 = E_1 + y_1(\alpha_1^{baru} - \alpha_1^{lama})K(\vec{x}_1, \vec{x}_2) + y_2(\alpha_2^{baru, potonga} - \alpha_2^{lama})K(\vec{x}_2, \vec{x}_2) + b^{lama} \dots\dots\dots(2-31)$$

dimana:

- $\alpha_i$  = *Lagrange Multipliers*



- $y_i$  = kelas (sampel positif (+1) dan sampel negatif(-1))
- $K(\vec{x}_1, \vec{x}_1)$  = fungsi kernel
- $b$  = bias
- $E_i$  = error pada data training ke- $i$

Nilai  $b_1$  dan  $b_2$  valid maka keduanya akan dipakai. Jika kedua pengali *Lagrange* yang baru berada pada batas dan jika  $L$  tidak sama dengan  $H$ , maka interval antara  $b_1$  dan  $b_2$  adalah semua bias yang konsisten dengan kondisi KKT. SMO akan memilih bias yang merupakan setengah dari nilai  $b_1$  dan  $b_2$

**2.8.1 Simplified Sequential Minimal Optimization (SSMO)**

*Simplified Sequential Minimal Optimization* (SSMO) merupakan algoritma yang dihasilkan dari modifikasi algoritma SMO. Pada algoritma ini memilih dua parameter  $\alpha$ , nilai objektif,  $\alpha_i$  dan  $\alpha_j$  akan dioptimalkan secara bersama-sama untuk kedua nilai alpha. Akhirnya menyesuaikan parameter  $b$  berdasarkan nilai  $\alpha$  yang baru, proses ini diulang sampai nilai  $\alpha$  konvergen[16: 1-5].

Pada algoritma SMO memilih secara heuristik untuk memilih  $\alpha_i$  dan  $\alpha_j$  yang akan dioptimalkan sehingga memaksimalkan fungsi objektif sebanyak mungkin. Pada jumlah data set yang besar, ini sangat penting untuk kecepatan dari proses algoritma, karena ada  $m(m - 1)$  kemungkinan untuk  $\alpha_i$  dan  $\alpha_j$  dan beberapa akan menghasilkan jauh lebih sedikit peningkatan dari yang lain. Algoritma *simplified SMO* akan menyederhanakan pemilihan secara heuristik pada  $\alpha$ . Algoritma ini akan melakukan iterasi untuk semua  $\alpha_i, i = 1, \dots, m$ . Jika  $\alpha_i$  memenuhi kondisi *Karush-Kuhn-Tucker* (KKT) dengan nilai toleransi tertentu maka memilih  $\alpha_j$  secara acak dari sisa  $m - 1$  dan mencoba mengoptimalkan secara bersama-sama  $\alpha_i$  dan  $\alpha_j$ . Jika  $\alpha$  tidak ada yang berubah dari beberapa iterasi maka algoritma berakhir.

Setelah memilih  $\alpha_i$  dan  $\alpha_j$  untuk dioptimalkan, menghitung batasan dari nilai parameter yang digunakan. Pertama mencari batasan pada  $L$  dan  $H$  sehingga  $L \leq \alpha_j \leq H$  untuk  $\alpha_j$  memenuhi batasan  $0 \leq \alpha_j \leq C$ . Hal ini dapat ditunjukkan pada persamaan (2-32) dan (2-33).

- Jika  $y_i \neq y_j, L = \max(0, \alpha_j - \alpha_i), H = \min(C, C + \alpha_j - \alpha_i)$ .....(2-32)

- Jika  $y_i = y_j, L = \max(0, \alpha_j + \alpha_i - C), H = \min(C, \alpha_j + \alpha_i)$ .....(2-33)

dimana:

$\alpha_i, \alpha_j$  = Lagrange Multipliers pada data ke-i dan ke-j

$y_i, y_j$  = kelas (sampel positif (+1) dan sampel negatif(-1))

$C$  = batasan *error* bernilai positif

Mencari nilai  $\alpha_j$  sehingga memaksimalkan fungsi objektif. Jika nilai ini diluar L dan H maka nilai  $\alpha_j$  berada pada rentangan ini. Persamaan untuk mencari  $\alpha_j$  ditunjukkan pada persamaan (2-34).

$$\alpha_j = \alpha_j - \frac{y_j(E_i - E_j)}{\eta} \dots \dots \dots (2-34)$$

dimana:

$$f(x) = \sum_{i=1}^n \alpha_i y_i K(x, x_i) + b \dots \dots \dots (2-35)$$

$$E_k = f(x) - y_k \dots \dots \dots (2-36)$$

$$\eta = 2K(x_i, x_j) - K(x_i, x_i) - K(x_j, x_j) \dots \dots \dots (2-37)$$

Selanjutnya memotong  $\alpha_j$  pada rentangan nilai L dan H dengan persamaan (2-38).

$$\alpha_j = \begin{cases} H, & \text{if } \alpha_j > H; \\ \alpha_j, & \text{if } L < \alpha_j < H; \\ L, & \text{if } \alpha_j < L; \end{cases} \dots \dots \dots (2-38)$$

Akhirnya, setelah menyelesaikan  $\alpha_j$  maka nilai akhir untuk  $\alpha_j$  ditunjukkan pada persamaan (2-39).

$$\alpha_i = \alpha_i + y_i * y_j (\alpha_j^{lama} - \alpha_j) \dots \dots \dots (2-39)$$

dimana:

$\alpha_j^{lama}$  = nilai dari  $\alpha_j$  sebelum dilakukan optimasi pada persamaan (2-34) dan (2-38).

Setelah mengoptimalkan  $\alpha_i$  dan  $\alpha_j$ , memilih b sedemikian hingga kondisi KKT terpenuhi pada data  $i$  dan  $j$ . Jika setelah optimasi,  $\alpha_i$  tidak berada pada batas  $0 \leq \alpha_i \leq C$  maka nilai  $b_1$  dapat dihitung dengan menggunakan persamaan (2-40). Jika  $\alpha_j$  tidak berada pada batas  $0 \leq \alpha_j \leq C$  maka nilai  $b_2$  dapat dihitung dengan menggunakan persamaan (2-41). Jika  $\alpha_i$  dan  $\alpha_j$  tidak memenuhi kedua kondisi tersebut maka nilai b didapatkan dengan  $b = (b_1 + b_2)/2$ .





$$b_1 = b - E_i - y_i(\alpha_i - \alpha_i^{lama})K(x_i, x_i) - y_j(\alpha_j - \alpha_j^{lama})K(x_i, x_j) \dots (2-40)$$

$$b_2 = b - E_j - y_i(\alpha_i - \alpha_i^{lama})K(\vec{x}_i, \vec{x}_j) - y_j(\alpha_j - \alpha_j^{lama})K(x_j, x_j) \dots (2-41)$$

$$b = \begin{cases} b_1, & \text{if } 0 < \alpha_i < C \\ b_2, & \text{if } 0 < \alpha_j < C \\ \frac{(b_1+b_2)}{2}, & \text{otherwise} \end{cases} \dots (2-42)$$

dimana:

$\alpha_i, \alpha_j$  = Lagrange Multipliers

$y_i, y_j$  = kelas (sampel positif (+1) dan sampel negatif (-1))

$K(\vec{x}_1, \vec{x}_1)$  = fungsi kernel

$b$  = bias

$E_i, E_j$  = error pada data training ke- $i$  dan data ke- $j$

Alur prosedur pada algoritma *Simplified SMO* sebagai berikut:

- 1) Inisialisasi awal  
Inisialisasi awal  $\alpha$ ,  $bias$  dan maksimum iterasi
- 2) Melakukan perhitungan nilai *error* pada data ke- $i$  untuk semua data latih dengan menggunakan persamaan (2-36) dan (2-35).
- 3) Melakukan pengecekan terhadap kondisi KKT dimana  $\alpha_i < C$  dan  $\alpha_i > 0$
- 4) Memilih data ke- $j$  sebagai data pembanding secara acak (*random*) yang akan dioptimasi secara bersama-sama dengan data ke- $i$ .
- 5) Menghitung nilai *error* pada data ke- $j$  dengan menggunakan persamaan (2-36) dan (2-35).
- 6) Menyimpan nilai  $\alpha_i$  dan  $\alpha_j$  awal ke dalam variabel  $\alpha_i^{lama}$  dan  $\alpha_j^{lama}$ .
- 7) Menghitung nilai L dan H menggunakan persamaan (2-32) dan (2-33) yang sangat bergantung pada target ( $y$ ) yang dimiliki pada tiap data latih.
- 8) Menghitung nilai eta ( $\eta$ ) sesuai dengan persamaan (2-37).
- 9) Menghitung nilai  $\alpha$  baru menggunakan persamaan (2-34) dan (2-38).
- 10) Menghitung nilai  $\alpha_i$  menggunakan persamaan (2-39).
- 11) Menghitung nilai bias berdasarkan kondisi yang terdapat pada persamaan (2-42).

- 12) Langkah-langkah diatas dilakukan sampai nilai  $\alpha$  pada semua data latih tidak memenuhi kondisi KKT dan tidak mengalami perubahan.

Pseudo-code <i>Simplified SMO</i>	
<b>Input:</b>	<p>C : nilai complexity            tol: nilai toleransi            max_iter: jumlah maksimum iterasi semua data latih tanpa adanya perubahan nilai <math>\alpha</math>  <math>(x_1, y_1), \dots, (x_m, y_m)</math>: data latih</p>
<b>Output:</b>	<p><math>\alpha</math> : untuk semua data latih            b : nilai bias</p>
	<ul style="list-style-type: none"> <li>• Inisialisasi awal <math>\alpha = 0</math>, <math>b = 0</math>, <math>pass = 0</math></li> <li>• while (<math>pass &lt; max\_iter</math>)               <ul style="list-style-type: none"> <li>• <math>alpha\_berubah = 0</math>.</li> <li>• for <math>i = 1</math> to <math>m</math> <ul style="list-style-type: none"> <li>• hitung <math>E_i = f(x_i) - y_i</math> menggunakan persamaan (2-36)</li> <li>• if (<math>(y_i E_i &lt; -tol \ \&amp;\&amp; \ \alpha_i &lt; C) \    \ (y_i E_i &gt; tol \ \&amp;\&amp; \ \alpha_i &gt; 0)</math>)                   <ul style="list-style-type: none"> <li>• memilih <math>j \neq i</math> (random)</li> <li>• hitung <math>E_j = f(x_j) - y_j</math> menggunakan persamaan (2-36)</li> <li>• simpan nilai <math>\alpha</math>: <math>\alpha_{i\_lama} = \alpha_i</math>, <math>\alpha_{j\_lama} = \alpha_j</math></li> <li>• hitung L dan H menggunakan persamaan (2-32) dan (2-33)</li> <li>• if (<math>L == H</math>)                       <ul style="list-style-type: none"> <li>• <b>continue</b> <math>i++</math>.</li> </ul> </li> <li>• hitung <math>\eta</math> menggunakan persamaan (2-37)</li> <li>• if (<math>\eta \geq 0</math>)                       <ul style="list-style-type: none"> <li>• <b>continue</b> <math>i++</math>.</li> </ul> </li> <li>• hitung <math>\alpha_j</math> baru menggunakan persamaan (2-34) dan (2-38)</li> <li>• if (<math> \alpha_j - \alpha_{j\_lama}  &lt; 10^{-5}</math>)                       <ul style="list-style-type: none"> <li>• <b>continue</b> <math>i++</math>.</li> </ul> </li> <li>• menentukan nilai <math>\alpha_i</math> menggunakan persamaan (2-39)</li> <li>• menghitung bias berdasarkan kondisi pada persamaan (2-42)</li> <li>• <math>alpha\_berubah = alpha\_berubah + 1</math></li> </ul> </li> <li>• if berakhir.</li> </ul> </li> <li>• for berakhir.</li> <li>• if (<math>alpha\_berubah == 0</math>)               <ul style="list-style-type: none"> <li>• <math>pass = pass + 1</math></li> </ul> </li> <li>• else               <ul style="list-style-type: none"> <li>• <math>pass = 0</math></li> </ul> </li> <li>• while berakhir.</li> </ul> </li></ul>

Gambar 2.12 Pseudocode *Simplified SMO*

Sumber: [16:4]

## 2.9 Rata-rata (Mean)

Rata rata adalah nilai khas yang mewakili sifat tengah, atau posisi pusat dari suatu kumpulan nilai data. Rumus untuk mencari nilai rata rata ditunjukkan pada persamaan (2-43) sebagai berikut.

$$\bar{X} = \frac{\sum_{i=1}^n X_{ij}}{n} \dots \dots \dots (2-43)$$

Dimana :

$\bar{X}$  = rata-rata (Mean)

$X_{ij}$  = nilai RGB dari baris ke  $i$  dan kolom ke  $j$

$n$  = banyaknya jumlah data RGB

Contoh kasus untuk menghitung rata-rata pada nilai RGB pada sebuah citra berwarna. Contoh kasus menggunakan sebuah citra berwarna dari hasil *cropping* citra Sapi Bali (SB-1) dengan ukuran  $32 \times 32$  pixel. Pada perhitungan untuk mencari rata-rata RGB menggunakan potongan  $3 \times 3$  pixel dari citra tersebut. Komponen warna RGB dari citra tersebut yang digunakan dalam perhitungan rata-rata RGB dapat dilihat pada tabel 2.11.

Tabel 2.11 Nilai RGB Pada Citra Berukuran  $3 \times 3$  Pixel

$X_{ij}$	1	2	3
1	R = 62 G = 62 B = 50	R = 62 G = 62 B = 50	R = 62 G = 62 B = 50
2	R = 63 G = 63 B = 51	R = 63 G = 63 B = 51	R = 63 G = 63 B = 51
3	R = 65 G = 65 B = 53	R = 65 G = 65 B = 53	R = 65 G = 65 B = 53

Sumber: [SB-2][Lampiran 4]

Pada tabel 2.11 dapat dilihat bahwa pixel pada baris ( $i$ ) ke-1 dan kolom ( $j$ ) ke-2 memiliki nilai R = 113, G = 111 dan B = 122. Untuk menghitung rata-rata nilai tiap R, G, dan B dapat dilihat pada perhitungan dibawah ini dengan mengacu pada persamaan (2-43).

$$\begin{aligned} \bar{X}_R &= \frac{X_{11} + X_{12} + X_{13} + X_{21} + \dots + X_{33}}{9} \\ &= \frac{62 + 62 + 62 + 63 + \dots + 65}{9} \\ &= \frac{570}{9} \end{aligned}$$

$$= 63,333333$$

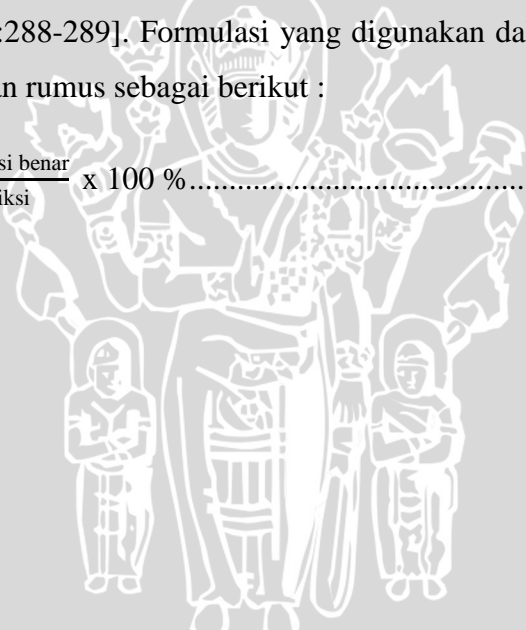
Dari perhitungan di atas didapatkan nilai rata-rata untuk komponen warna R (*red*) sebesar 63,333333. Untuk mencari nilai rata-rata komponen warna G (*green*) dan B (*blue*) dapat menggunakan cara yang sama pada perhitungan nilai rata-rata komponen warna R (*red*).

## 2.10 Evaluasi

Tahap evaluasi pada penelitian ini adalah bertujuan untuk mengetahui tingkat akurasi dari penggunaan algoritma *Simplified* SMO. Perhitungan pada tahap ini dengan cara menghitung jumlah data uji yang kelasnya diprediksi secara benar dibandingkan dengan data uji yang kelasnya diprediksi salah.

Pengukuran akurasi dan ketepatan suatu model dalam sistem dapat dilakukan dengan cara menghitung perbandingan jumlah prediksi benar terhadap seluruh *record* yang dapat diprediksi [13:288-289]. Formulasi yang digunakan dalam evaluasi pada penelitian ini menggunakan rumus sebagai berikut :

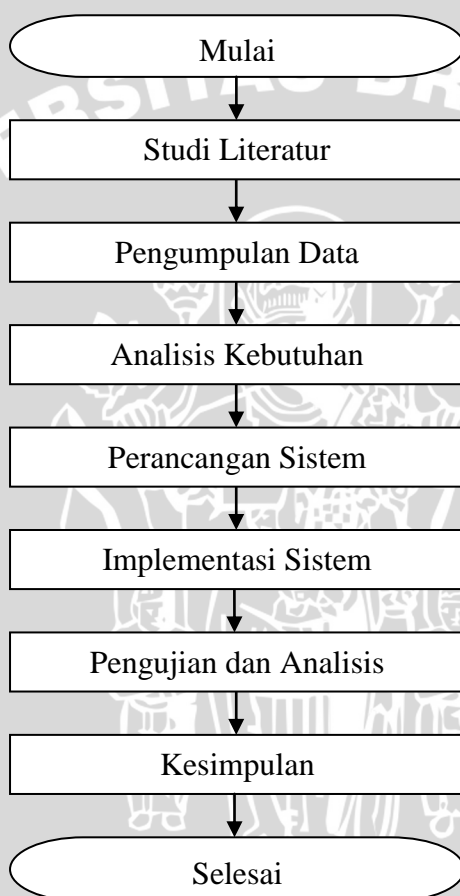
$$\text{Akurasi} = \frac{\text{jumlah prediksi benar}}{\text{total prediksi}} \times 100 \% \dots\dots\dots(2-44)$$



### BAB III

#### METODOLOGI PENELITIAN

Bab ini menjelaskan langkah-langkah yang akan dilakukan dalam penelitian. Penelitian dilakukan dengan melakukan tahapan-tahapan yaitu studi literatur, pengumpulan data, analisis kebutuhan dan perancangan, implementasi serta pengujian dari perangkat lunak yang akan dibuat. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan selanjutnya.



Gambar 3.1 Diagram Alir Metodologi Penelitian  
Sumber: [Perancangan]

#### 3.1 Studi Literatur

Tahap ini dilakukan untuk mendapatkan dasar teori dan sumber acuan untuk pembuatan sistem pendukung keputusan pemilihan bibit unggul Sapi Bali berdasarkan warna kulit menggunakan algoritma *Simplified SMO* diantaranya:

1. Sapi Bali

2. Metode SVM
3. Algoritma *Simplified* SMO
4. Citra Digital

Literatur tersebut diperoleh dari buku, jurnal, *ebook*, penelitian sebelumnya dan dokumentasi project.

### 3.2 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. Data-data tersebut digunakan untuk mengukur tingkat akurasi pada tahap pengujian sistem.

Pengumpulan data pada penelitian ini menggunakan data primer. Data primer merupakan data yang didapatkan secara langsung dari sumber penelitian. Data yang didapatkan berupa citra Sapi Bali yang memiliki kategori dan jenis kelamin yang berbeda.

### 3.3 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan kebutuhan yang diperlukan untuk membangun sistem pendukung keputusan. Analisis kebutuhan diterapkan sesuai dengan lokasi penelitian, variabel penelitian dan mempersiapkan kebutuhan penelitian. Secara keseluruhan, kebutuhan yang digunakan dalam pembuatan SPK ini meliputi:

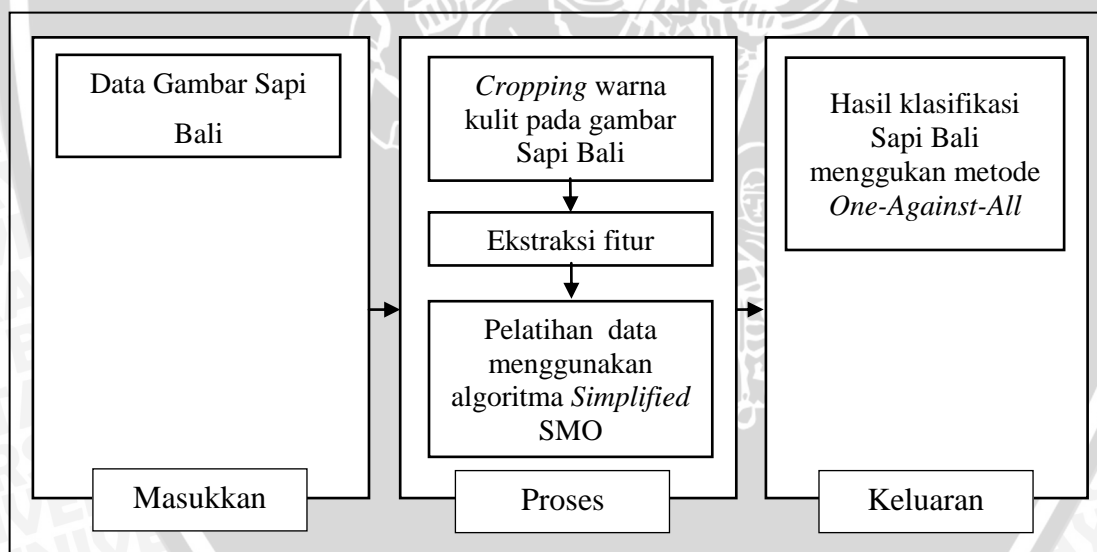
1. Spesifikasi kebutuhan *hardware*, meliputi:
  - Laptop dengan processor Intel(R) Core (TM) i3 CPU M 330 @ 2.13GHz (4CPUs)
  - Memory 2GB
2. Spesifikasi kebutuhan *software*, meliputi:
  - Windows 7 Home Premium 32-bit.
  - XAMPP.
  - NetBeans IDE 7.3.1.
  - *Java Development Kit* (JDK) 7.
3. Data yang dibutuhkan, meliputi:
  - Citra Sapi Bali beserta kelasnya.

### 3.4 Perancangan Sistem

Perancangan sistem dibuat berdasarkan hasil pengambilan data dan analisis kebutuhan yang dilakukan. Perancangan sistem dilakukan agar pengimplementasian sistem menjadi lebih mudah.

#### 3.4.1. Diagram Blok Sistem

Diagram blok sistem menjelaskan cara kerja sistem yang akan dibuat mulai dari masukan sampai keluaran yang dihasilkan oleh sistem. Secara umum sistem yang akan dibangun menggunakan algoritma *Simplified SMO* yang bertujuan untuk mengklasifikasikan kualitas Sapi Bali berdasarkan warna kulit. Sistem menyimpan *dataset* citra Sapi Bali yang dimasukkan oleh petugas BPTU (*admin*) berupa nilai RGB yang didapatkan dari proses ekstraksi fitur. *Dataset* tersebut akan digunakan untuk melakukan proses skenario uji coba yang dilakukan oleh peternak guna mendapatkan hasil akurasi yang terbaik dari masing-masing skenario. Gambar 3.2 merupakan diagram blok yang menggambarkan secara umum dari sistem yang akan dibangun.

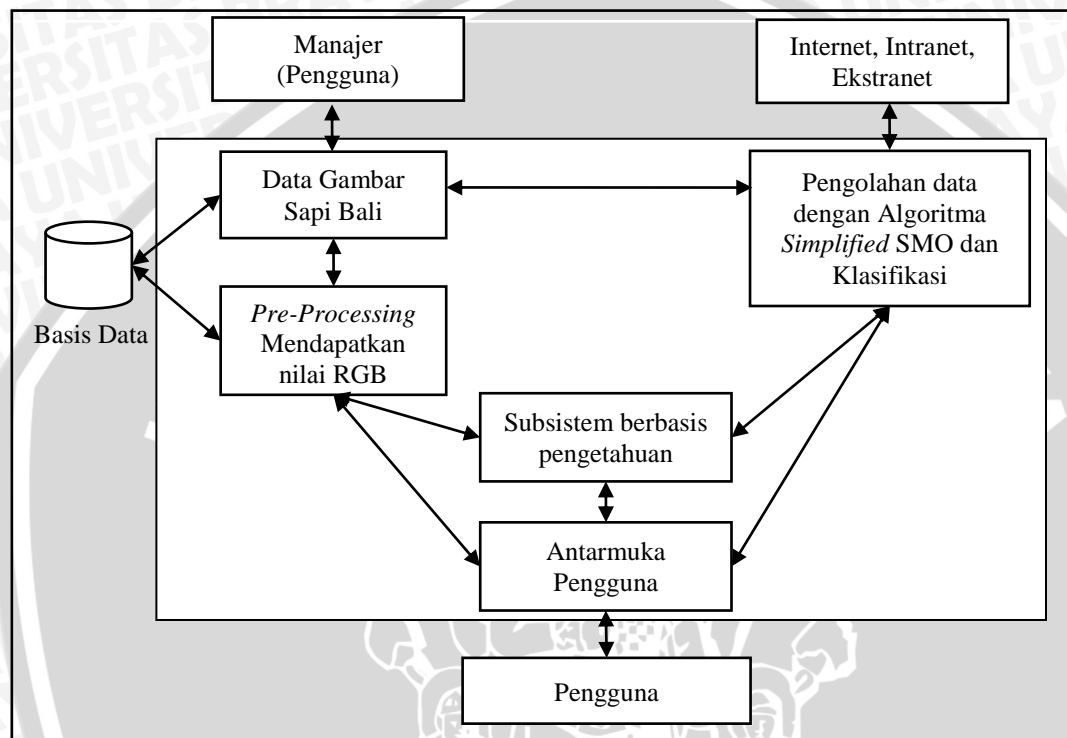


Gambar 3.2 Diagram Blok SPK Pemilihan Bibit Unggul Sapi Bali Menggunakan Algoritma *Simplified SMO*  
Sumber: [Perancangan]

#### 3.4.2. Arsitektur Sistem

Sistem yang akan dibuat untuk pendukung keputusan pemilihan bibit unggul Sapi Bali menerapkan algoritma *Simplified SMO*. Sistem ini dibagi menjadi dua bagian utama yaitu sistem untuk petugas BPTU (*admin*) dan sistem untuk peternak

(user). Pada sistem petugas BPTU berfungsi untuk memasukkan *dataset* ke dalam *database* sedangkan sistem peternak berfungsi untuk melakukan skenario uji coba pada data uji. Sistem akan memberikan keluaran berupa hasil klasifikasi (Baik, Sedang atau Buruk) dan akurasi yang didapatkan dari masing-masing skenario. Perancangan sistem ini dapat dilihat lebih jelas pada arsitektur sistem pendukung keputusan pada Gambar 3.3.



Gambar 3.3 Arsitektur SPK Pemilihan Bibit Unggul Sapi Bali Menggunakan Algoritma *Simplified SMO*  
Sumber: [Perancangan]

### 3.5 Implementasi Sistem

Implementasi perangkat lunak yang menerapkan algoritma *Simplified SMO* dilakukan dengan mengacu pada perancangan sistem. Implementasi perangkat lunak ini menggunakan bahasa pemrograman *Java*, database *MySQL* dan *tools* pendukung lainnya. Implementasi sistem tersebut meliputi:

- Pembuatan antar muka.
- Memasukkan data citra Sapi Bali ke *database MySQL* untuk diolah menjadi informasi yang berguna bagi sistem.
- Menerapkan algoritma *Simplified SMO* dalam aplikasi yang telah dibuat dengan bahasa *Java*.



- Output berupa klasifikasi kelas Baik (Bibit Unggul), Sedang atau Buruk (Penyimpangan).

### 3.6 Pengujian dan Analisis

Pengujian sistem ini dilakukan agar dapat memastikan bahwa aplikasi yang telah dibangun dapat bekerja sesuai dengan spesifikasi dari kebutuhan yang telah ditentukan. Pengujian yang dilakukan adalah pengujian tingkat akurasi data dengan membandingkan keluaran dari sistem dengan hasil pakar menggunakan algoritma *Simplified SMO*. Setiap skenario pengujian akan dilakukan sebanyak lima kali percobaan. Terdapat beberapa skenario pengujian yang dapat dilakukan yaitu sebagai berikut:

1. Pengujian untuk mengetahui pengaruh ukuran *cropping* dan rasio perbandingan jumlah data latih dan data uji terhadap hasil akurasi. Pengujian tersebut menggunakan lima tipe rasio perbandingan data latih dan data uji yaitu 90:10, 80:20, 70:30, 60:40 dan 50:50. Pengambilan data latih dan data uji dilakukan secara acak (*random*). Pengujian rasio tersebut akan dilakukan terhadap tiga buah ukuran *cropping* yaitu  $32 \times 32$  pixel,  $64 \times 64$  pixel, dan  $128 \times 128$  pixel.
2. Pengujian untuk mengetahui pengaruh nilai parameter *C* terhadap hasil akurasi. Pengujian ini dilakukan menggunakan rasio dan ukuran *cropping* yang memiliki tingkat akurasi terbaik terhadap lima nilai *C* yang berbeda yaitu 1, 4, 16, 64, dan 256.
3. Pengujian untuk mengetahui pengaruh nilai parameter *gamma* ( $\gamma$ ) terhadap hasil akurasi. Pengujian ini dilakukan menggunakan tingkat akurasi terbaik pada pengujian ke 2. Nilai *gamma* yang digunakan pada pengujian ini yaitu 0.0001, 0.001, 0.01, 0.1 dan 1.
4. Pengujian untuk mengetahui pengaruh jumlah iterasi terhadap hasil akurasi. Jumlah iterasi yang digunakan pada pengujian ini yaitu 1, 10, 100, 1000 dan 10000.
5. Pengujian terhadap 15 data uji untuk tiga kelas dan jenis kelamin

### 3.7 Kesimpulan

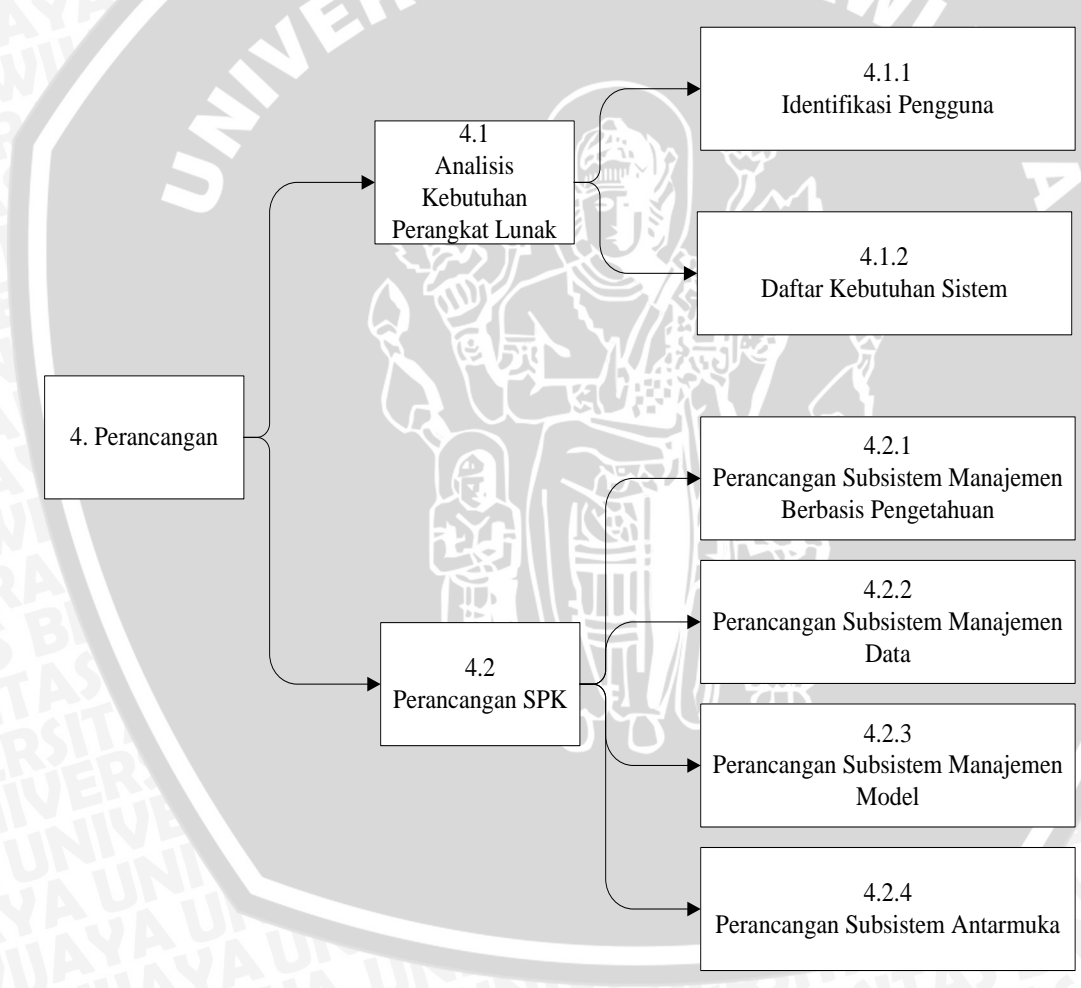
Kesimpulan dilakukan setelah semua tahapan, perancangan, implementasi algoritma *Simplified SMO*, dan pengujian sistem terhadap algoritma yang digunakan.

Penulisan saran berguna untuk memperbaiki kesalahan-kesalahan yang terjadi serta untuk memberikan pertimbangan atas pengembangan metode dan aplikasi selanjutnya.



## BAB IV PERANCANGAN

Bab ini menjelaskan tentang analisis kebutuhan perangkat lunak dan perancangan sistem pendukung keputusan Analisis kebutuhan perangkat lunak terdiri dari identifikasi aktor dan daftar kebutuhan sistem. Perancangan SPK ini terdiri dari perancangan subsistem manajemen berbasis pengetahuan, subsistem manajemen data, subsistem manajemen model, dan subsistem antarmuka. Pada gambar 4.1 menunjukkan bahwa pohon perancangan sebagai gambaran umum pokok bahasan pada bab 4.



Gambar 4.1 Pohon Perancangan  
Sumber :[Perancangan]



#### 4.1 Analisis Kebutuhan Perangkat Lunak

Sistem pemilihan bibit unggul Sapi Bali adalah sebuah usaha untuk meningkatkan kualitas dan pencegahan generasi bibit unggul Sapi Bali dari kepunahan. Permasalahan utama dari pemilihan bibit unggul Sapi Bali ini terletak pada metode dalam pemilihan bibit unggul Sapi Bali yang masih dilakukan secara manual. Pemilihan dilakukan dengan melihat secara langsung dengan melihat pola warna Sapi Bali yang akan diklasifikasikan ke dalam kelas yang telah ditentukan.

Metode yang selama ini digunakan dalam pemilihan bibit unggul sering kali terdapat kesalahan yang dilakukan oleh peternak (*human error*) dalam pengklasifikasian dan pemilihan bibit unggul Sapi. Sistem yang akan dibangun ini berusaha mengatasi permasalahan tersebut dengan membuat sistem yang berbasis desktop yang dapat digunakan oleh peternak di BPTU Sapi Bali. Sistem ini diharapkan dapat mengurangi tingkat kesalahan pengklasifikasian dan pemilihan bibit unggul Sapi Bali. Pada analisis kebutuhan ini diawali dengan mengidentifikasi aktor yang terlibat dengan sistem dan penjabaran kebutuhan fungsional. Analisis kebutuhan ini ditujukan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

##### 4.1.1 Identifikasi Pengguna

Pada penelitian ini terdapat 2 pengguna yang akan berinteraksi dengan sistem yaitu petugas BPTU (*admin*) dan peternak (*user*). Petugas dapat melakukan *insert*, *update*, dan *delete* untuk data Sapi Bali dan *user* sesuai dengan kebutuhan. Peternak dapat melakukan *training* pada data latih dan skenario uji coba untuk mendapatkan tingkat akurasi. Tabel 4.1 menampilkan identifikasi dari kedua pengguna tersebut yang merupakan hasil dari proses identifikasi aktor.

Tabel 4.1 Identifikasi Pengguna

Pengguna	Deskripsi Pengguna	Keterangan
Petugas BPTU (PTGS)	Petugas BPTU merupakan aktor pengguna sistem yang bertugas untuk mengolah data <i>user</i> dan data Sapi Bali.	<i>Admin</i>
Peternak (PTK)	Peternak yang menggunakan sistem pendukung keputusan untuk melakukan skenario uji coba pada data Sapi Bali untuk diketahui tingkat akurasi.	<i>User</i>

Sumber: [Perancangan]

#### 4.1.2 Daftar Kebutuhan Sistem

Pada daftar kebutuhan sistem ini akan membahas tentang kebutuhan fungsional dan pengguna yang terlibat di dalamnya. Daftar kebutuhan fungsional kebutuhan sistem ditunjukkan pada Tabel 4.2.

Tabel 4.2 Daftar Kebutuhan Fungsional

Pengguna	Kebutuhan	Nama Aliran Data
PTGS PTK	Sistem menyediakan antarmuka untuk melakukan <i>Login</i>	<i>Login</i>
PTGS	Sistem menyediakan antarmuka untuk olah data Sapi Bali	Olah Data Sapi Bali
PTGS	Sistem menyediakan antarmuka untuk melihat secara detail informasi dari data Sapi Bali	Lihat Detail Sapi Bali
PTGS	Sistem menyediakan menu untuk menambah data Sapi Bali	Tambah Data Sapi Bali
PTGS	Sistem menyediakan menu untuk mengubah data Sapi Bali	Ubah Data Sapi Bali
PTGS	Sistem menyediakan menu untuk menghapus data Sapi Bali	Hapus Data Sapi Bali
PTGS	Sistem menyediakan menu untuk mengcropp citra Sapi Bali	<i>Cropp</i> Gambar Sapi Bali
PTGS	Sistem menyediakan antarmuka untuk olah data pengguna	Olah Data Pengguna
PTGS	Sistem menyediakan antarmuka untuk menambah pengguna baru	Tambah Data Pengguna
PTGS	Sistem menyediakan antarmuka untuk mengubah data pengguna	Ubah Data Pengguna
PTGS	Sistem menyediakan antarmuka untuk menghapus data pengguna	Hapus Data Pengguna
PTK	Sistem menyediakan menu untuk mengubah data peternak	Update Data Peternak
PTK	Sistem menyediakan menu untuk skenario uji coba data	Skenario Uji Coba Data
PTK	Sistem menyediakan menu untuk menampilkan hasil klasifikasi	Hasil Klasifikasi
PTK	Sistem menyediakan antarmuka untuk menampilkan hasil akurasi	Hasil Akurasi
PTGS PTK	Sistem menyediakan antarmuka untuk melakukan <i>Logout</i>	<i>Logout</i>

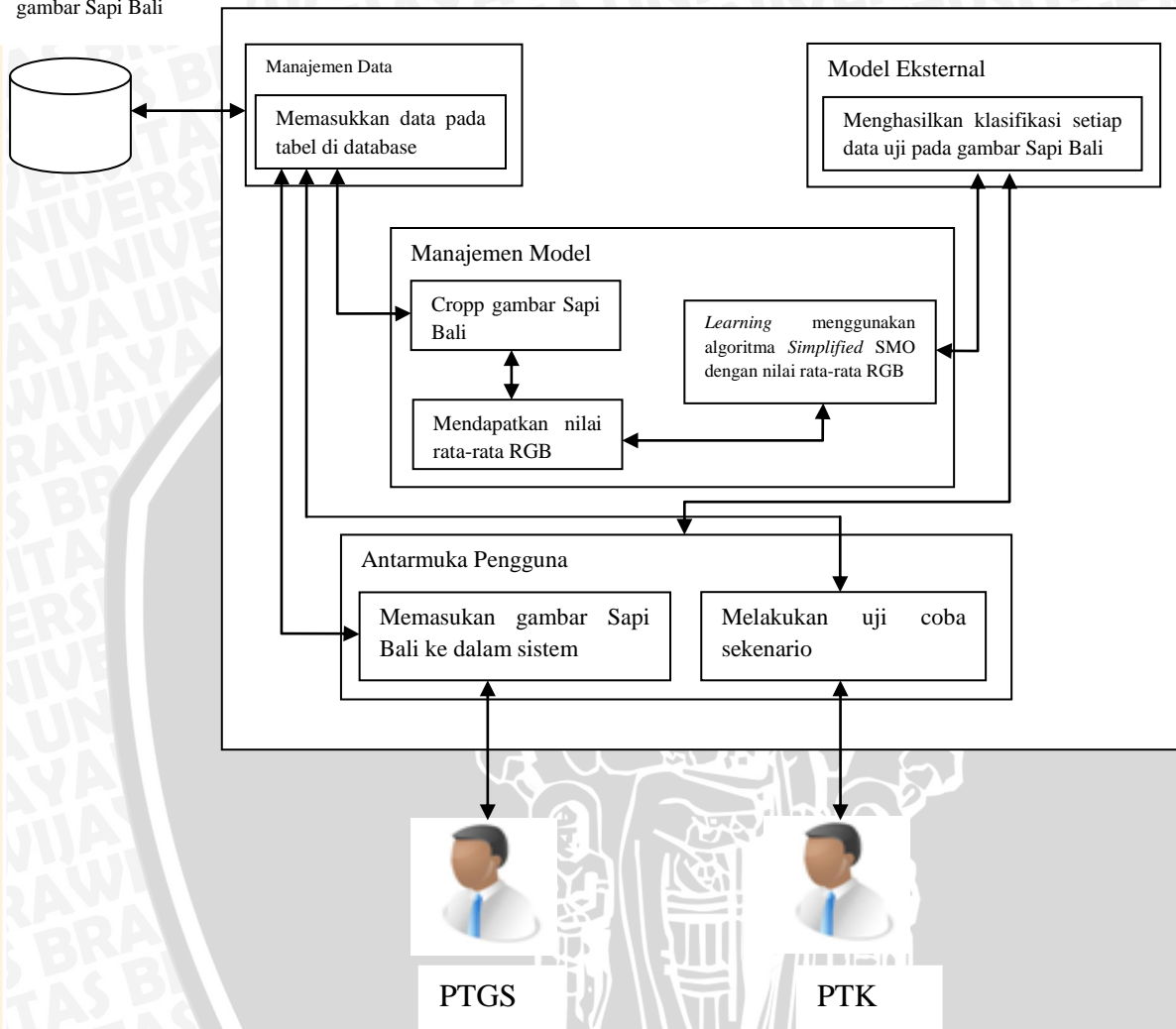
Sumber: [Perancangan]

#### 4.2 Perancangan SPK

Perancangan SPK terdiri dari perancangan subsistem manajemen berbasis pengetahuan, subsistem manajemen model, subsistem manajemen data, dan subsistem antarmuka pengguna. Arsitektur sistem dibangun mengacu pada

permodelan menggunakan algoritma *Simplified* SMO. Gambar 4.2 merupakan arsitektur SPK untuk pemilihan bibit unggul Sapi Bali.

Data eksternal & internal gambar Sapi Bali

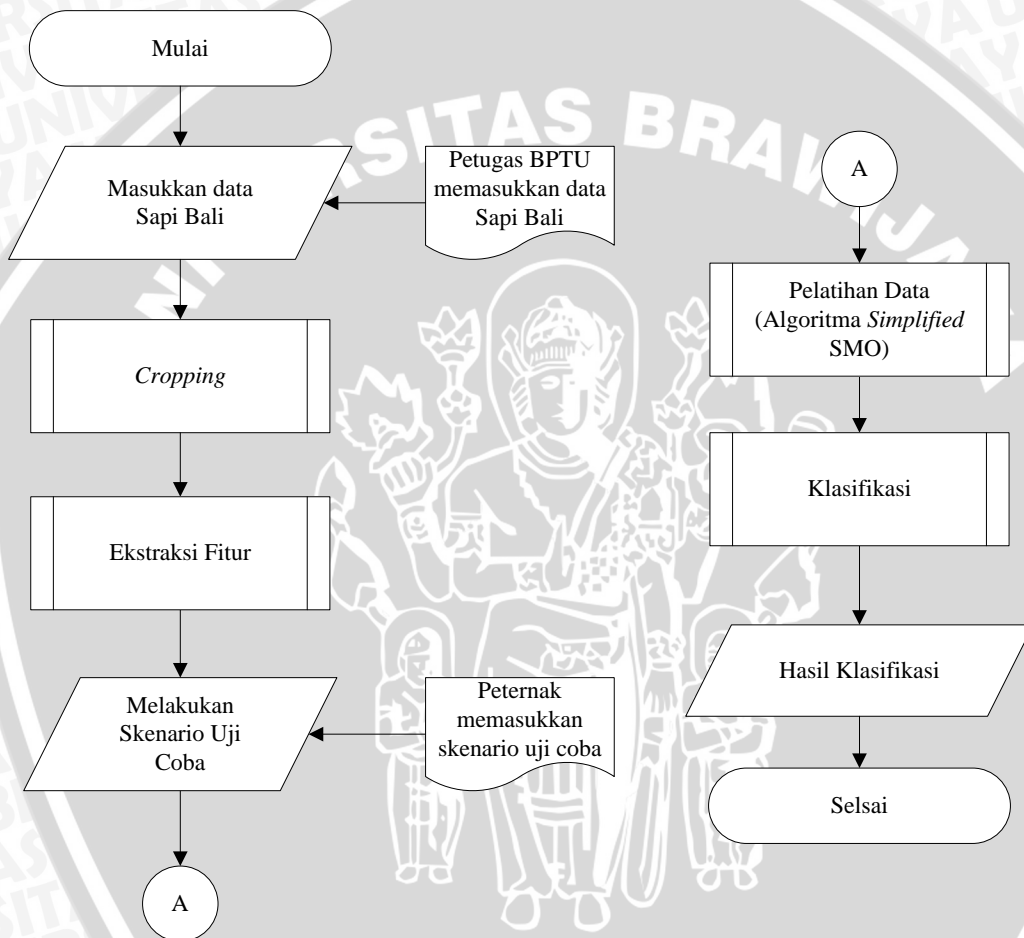


Gambar 4.2 Arsitektur SPK pemilihan bibit unggul Sapi Bali  
Sumber: [Perancangan]

Tahap umum SPK pemilihan bibit unggul Sapi Bali melakukan proses memasukkan *dataset* berupa citra 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. Hasil *cropping* citra Sapi Bali tersebut akan digunakan untuk melakukan skenario uji coba.

Tahap selanjutnya merupakan tahapan yang dilakukan oleh peternak. 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* citra Sapi Bali dan akan menghasilkan nilai optimal dari  $\alpha$  dan *bias* pada tiap data latih serta fungsi keputusan untuk proses klasifikasi. Tahap terakhir dilakukan proses klasifikasi menggunakan metode *one-against-all* untuk menangani klasifikasi *multi-class*. Diagram alir SPK pemilihan bibit unggul Sapi Bali menggunakan algoritma *Simplified SMO* dapat dilihat pada gambar 4.3.



Gambar 4.3 Diagram Alir SPK Pemilihan Bibit Unggul Sapi Bali  
 Sumber : [Perancangan]

#### 4.2.1 Subsistem Manajemen Berbasis Pengetahuan

Perancangan subsistem manajemen berbasis pengetahuan berisi tentang pengetahuan yang relevan dalam proses pemilihan bibit unggul Sapi Bali. Subsistem ini merupakan inti program dari SPK pemilihan bibit unggul Sapi Bali dimana basis pengetahuan ini merupakan representasi pengetahuan dari observasi langsung yang dilakukan di BPTU Sapi Bali.

Observasi yang dilakukan di BPTU untuk mendapatkan data citra Sapi Bali yang akan digunakan dalam proses pengujian. Penentuan tiap kelas pada data Sapi Bali dilakukan oleh pakar, dimana pakar yang digunakan dalam penelitian ini adalah peternak yang bekerja di BPTU. Terdapat 3 kelas yang digunakan untuk klasifikasi kualitas Sapi Bali yaitu baik, sedang dan buruk. Berikut adalah contoh keputusan pakar dalam menentukan klasifikasi kualitas Sapi Bali berdasarkan warna kulitnya.

Berdasarkan hasil wawancara terhadap pakar didapatkan bahwa klasifikasi kualitas Sapi Bali yang memiliki kategori “Baik” untuk Sapi Bali jantan dan betina ditunjukkan pada gambar 4.4 dan 4.5. Dimana warna kulit Sapi Bali jantan yang berkualitas baik adalah dominan berwarna hitam, sedangkan pada Sapi Bali betina yang berkualitas baik adalah coklat kemerahan.



Gambar 4.4 Sapi Bali Jantan Kualitas Baik  
Sumber : [Observasi]



Gambar 4.5 Sapi Bali Betina Kualitas Baik  
Sumber : [Observasi]



Berdasarkan hasil wawancara terhadap pakar didapatkan bahwa klasifikasi kualitas Sapi Bali yang memiliki kategori “Sedang” untuk Sapi Bali jantan dan betina ditunjukkan pada gambar 4.6 dan 4.7. Dimana warna kulit Sapi Bali jantan yang berkualitas sedang adalah sebagian hitam, sedangkan pada Sapi Bali betina yang berkualitas sedang adalah coklat.



Gambar 4.6 Sapi Bali Jantan Kualitas Sedang  
Sumber : [Observasi]



Gambar 4.7 Sapi Bali Betina Kualitas Sedang  
Sumber : [Observasi]

Berdasarkan hasil wawancara terhadap pakar didapatkan bahwa klasifikasi kualitas Sapi Bali yang memiliki kategori “Buruk” untuk Sapi Bali jantan dan betina ditunjukkan pada gambar 4.8 dan 4.9. Dimana warna kulit Sapi Bali jantan yang berkualitas buruk adalah coklat, sedangkan pada Sapi Bali betina yang berkualitas buruk adalah coklat pucat.



Gambar 4.8 Sapi Bali Jantan Kualitas Buruk  
Sumber : [Observasi]



Gambar 4.9 Sapi Bali Betina Kualitas Buruk  
Sumber : [Observasi]

Data citra Sapi Bali yang didapatkan pada observasi tidak semua memenuhi kebutuhan untuk dilakukan proses pemotongan (*cropping*) warna kulit pada bagian samping dari citra Sapi Bali. Hal tersebut terjadi dikarenakan pengambilan foto Sapi Bali memiliki jarak yang berbeda-beda, sehingga perlu dilakukan perbesaran objek (Sapi Bali) dengan bantuan aplikasi *photoshop*. Proses ini dilakukan agar pada saat pengambilan (*cropping*) sampel warna kulit pada Sapi Bali dapat memenuhi kebutuhan yang telah ditentukan. Berikut gambar 4.10 merupakan citra Sapi Bali yang memiliki objek yang kecil dan gambar 4.11 merupakan hasil perbesaran menggunakan aplikasi *photoshop*.



Gambar 4.10 Sapi Bali Betina  
Sumber : [Observasi]



Gambar 4.11 Sapi Bali Betina Setelah Dilakukan Proses Perbesaran  
Sumber : [Observasi]

Fitur yang digunakan dalam penelitian ini adalah nilai rata-rata dari tiga komponen warna yaitu *red*, *green* dan *blue* (RGB). Nilai rata-rata tersebut didapatkan dari pengambilan (*cropping*) sampel warna kulit pada citra Sapi Bali sebanyak tiga kali di area yang tidak sama (daerah kaki depan sampai leher, bagian perut dan bagian pinggul dari Sapi Bali). *Cropping* warna kulit Sapi Bali berbentuk persegi dengan ukuran  $32 \times 32$  pixel,  $64 \times 64$  pixel dan  $128 \times 128$  pixel. Berikut hasil *cropping* sampel warna kulit Sapi Bali dengan ukuran *cropping*  $128 \times 128$  pixel dapat dilihat pada tabel 4.3.

Tabel 4.3 Kriteria Fitur Warna Kulit Sapi Bali

Jenis Kelamin	Umur	Klasifikasi	Crop 1	Crop 2	Crop 3
Jantan	2 Tahun	Baik (SB-6)			
			R:72;G:65;B:75	R:104;G:93;B:100	R:79;G:68;B:75
		Sedang (SB-20)			
			R:105;G:73;B:49	R:122;G:88;B:61	R:144;G:107;B:78
		Buruk (SB-42)			
			R:185;G:167;B:144	R:196;G:175;B:151	R:182;G:158;B:136
Betina	1.5 Tahun	Baik (SB-46)			
			R:154;G:116;B:94	R:162;G:122;B:101	R:144;G:110;B:93
		Sedang (SB-64)			
			R:190;G:134;B:80	R:190;G:134;B:77	R:201;G:137;B:80
		Buruk (SB-86)			
			R:182;G:154;B:135	R:191;G:164;B:145	R:171;G:147;B:129

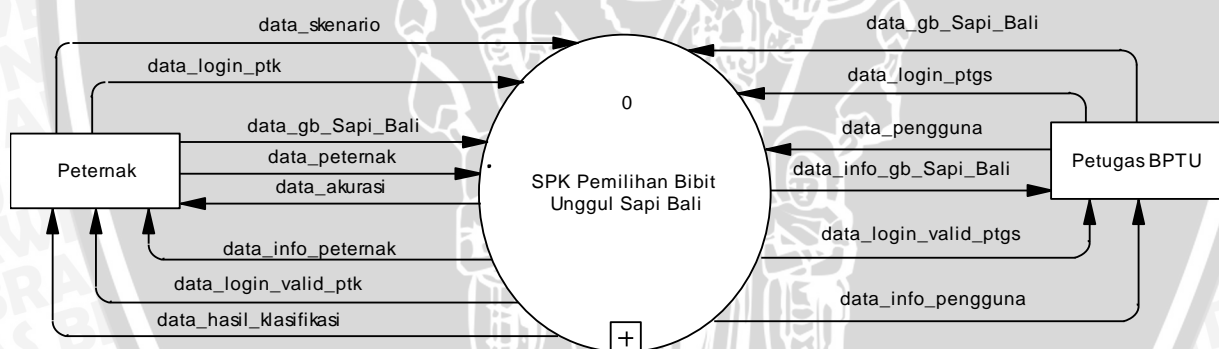
Sumber: [Observasi][Lampiran 4]

#### 4.2.2 Subsistem Manajemen Data

Subsistem manajemen data membahas tentang *Data Flow Diagram (DFD)*, *Entity Relation Diagram (ERD)* dan *Physical Diagram*.

##### 4.2.2.1 Data Flow Diagram (DFD)

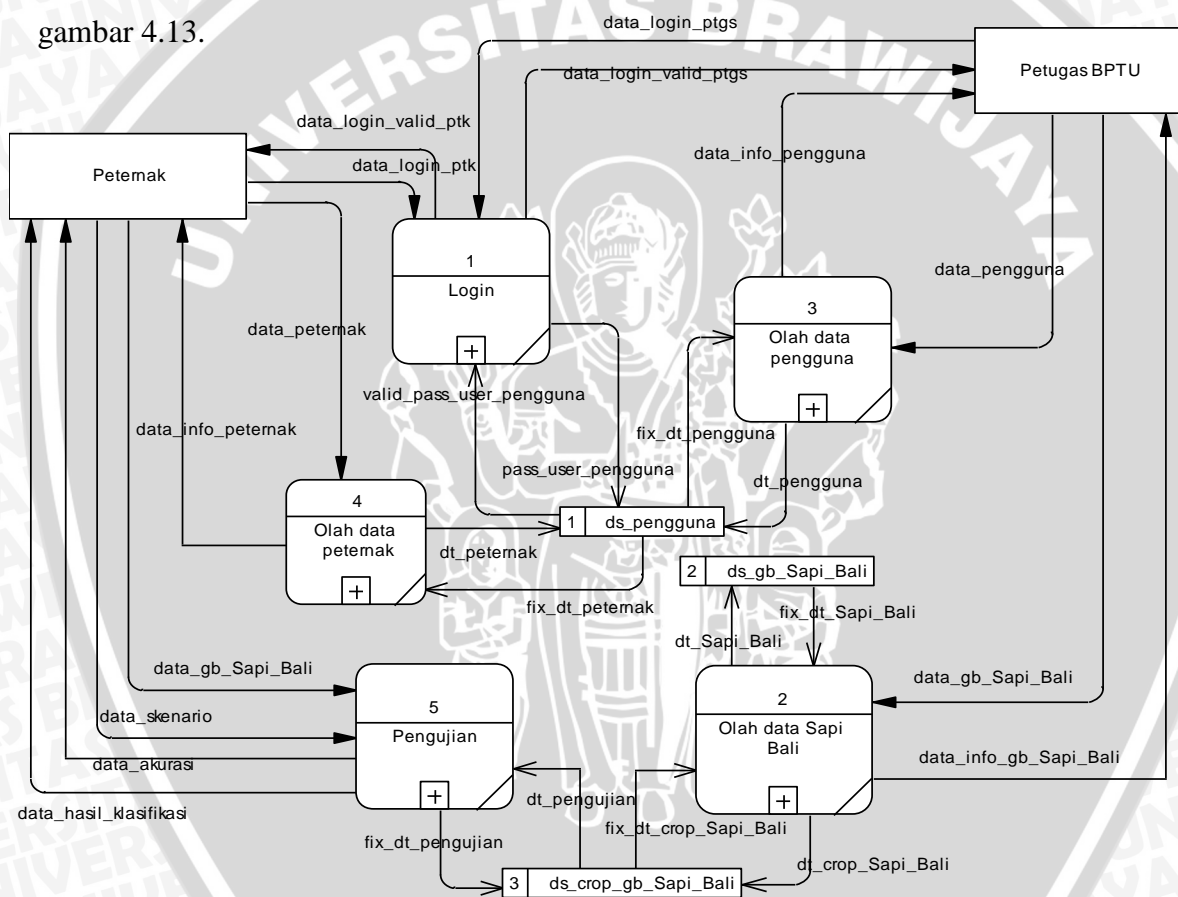
Pembuatan DFD bertujuan untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data. Diagram awal dari DFD dinamakan diagram konteks tetapi ada beberapa sumber yang menamakan diagram level 0 atau diagram level 1. Diagram konteks pada penelitian ini terdapat dua pengguna yaitu peternak dan petugas BPTU. Pengguna dapat melakukan proses login kedalam sistem dengan memasukkan *username* dan *password*. Sistem akan memberikan hak akses kepada pengguna berdasarkan *username* dan *password*. Petugas dapat melakukan pengisian *dataset* Sapi Bali dan data pengguna ke dalam database. Peternak dapat melakukan pengujian untuk mengetahui tingkat akurasi yang dihasilkan oleh sistem. Gambar 4.12 merupakan Diagram konteks pada penelitian ini.



Gambar 4.12 Diagram Konteks  
Sumber : [Perancangan]

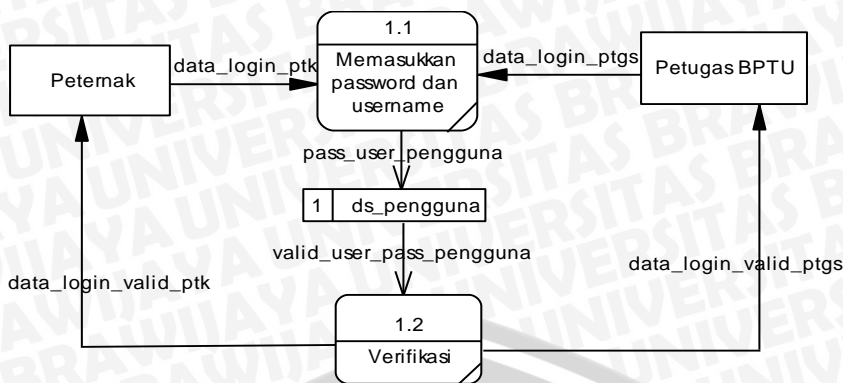
Diagram konteks akan didekomposisi menjadi beberapa sub proses diagram level selanjutnya. DFD level 1 merupakan dekomposisi dari diagram konteks. Pada diagram level 1 terdapat beberapa proses antara lain login, olah data Sapi Bali, olah data pengguna, olah data peternak dan pengujian. Terdapat data store yang digunakan untuk menyimpan data yang dibutuhkan antara lain *data store* pengguna, *data store* citra Sapi Bali dan *data store* crop citra Sapi Bali. Proses pertama adalah proses login. Proses ini dapat dilakukan oleh peternak dan petugas BPTU dengan memasukkan *username* dan *password* kemudian sistem akan memverifikasi

username dan password pada *database*. Proses kedua adalah proses olah data Sapi Bali. Proses ini dapat dilakukan oleh petugas BPTU untuk memasukkan data Sapi Bali sesuai kebutuhan ke dalam *database* yang akan dijadikan *dataset*. Proses ketiga adalah proses olah data pengguna. Proses ini dapat dilakukan oleh petugas BPTU untuk menambah, mengubah dan menghapus pengguna sistem. Proses keempat adalah olah data peternak. Proses ini hanya dapat dilakukan oleh peternak untuk mengubah atribut yang dimilikinya. Proses kelima adalah pengujian. Proses ini dilakukan oleh peternak untuk mengetahui tingkat akurasi yang diberikan oleh sistem. DFD level 1 dari SPK pemilihan bibit unggul Sapi Bali ditunjukkan pada gambar 4.13.



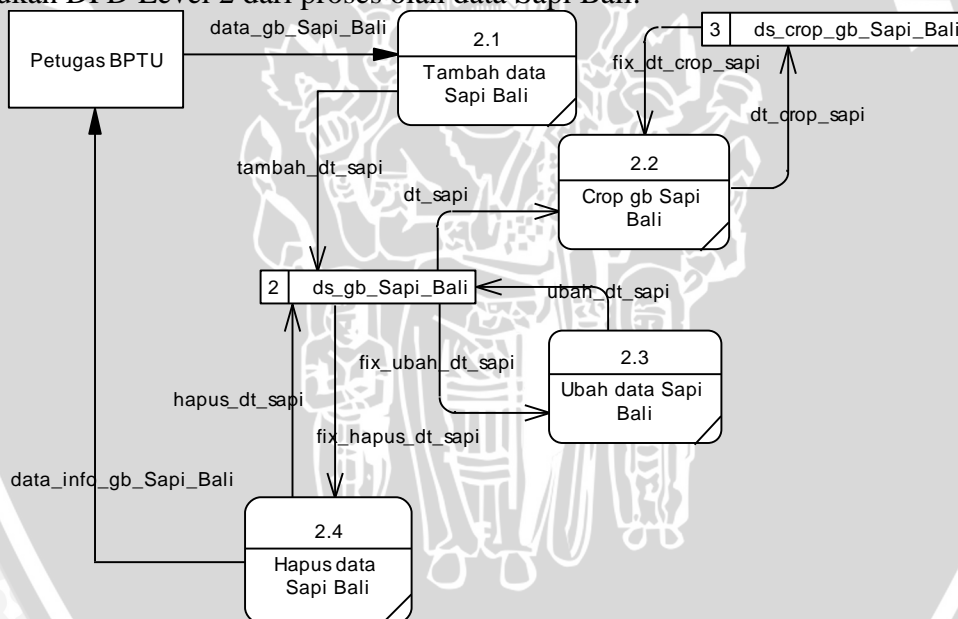
Gambar 4.13 DFD Level 1  
Sumber : [Perancangan]

Gambar 4.14 merupakan DFD level 2 dari proses *login*. Pada proses ini peternak dan petugas BPTU dapat masuk ke dalam sistem dengan menginputkan *username* dan *password* yang telah dimiliki. Sistem akan melakukan verifikasi terhadap inputan *username* dan *password* yang telah dilakukan oleh peternak dan petugas BPTU terhadap data yang tersimpan di *database*.



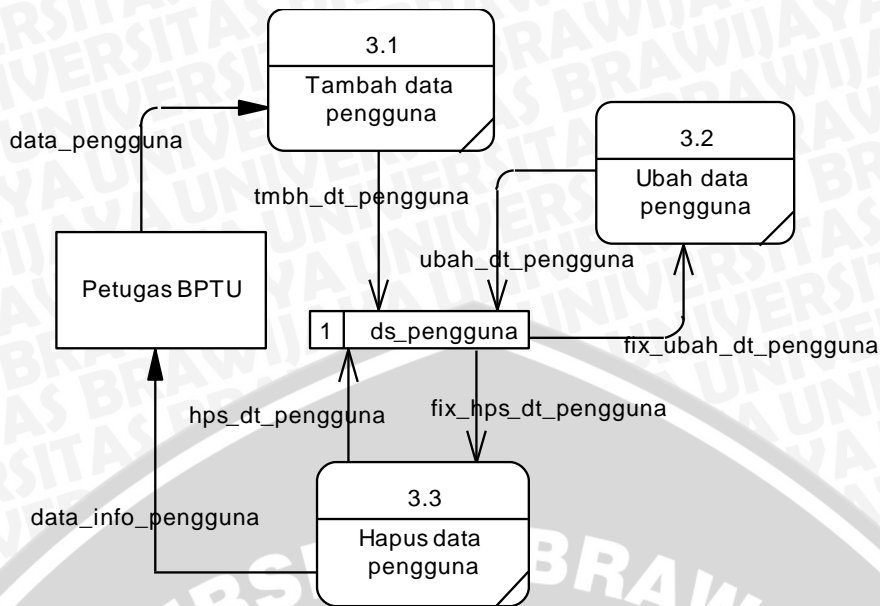
Gambar 4.14 DFD Level 2 Login  
 Sumber : [Perancangan]

Proses olah data Sapi Bali didekomposisi menjadi beberapa subproses antara lain tambah data Sapi Bali, crop gambar Sapi Bali, ubah data Sapi Bali, dan ubah data Sapi Bali. Pada subproses ini petugas BPTU dapat melakukan penambahan data Sapi Bali ke dalam sistem yang nantinya akan digunakan untuk proses skenario uji coba, mengubah data Sapi Bali dan menghapus data Sapi Bali. Gambar 4.15 menunjukkan DFD Level 2 dari proses olah data Sapi Bali.



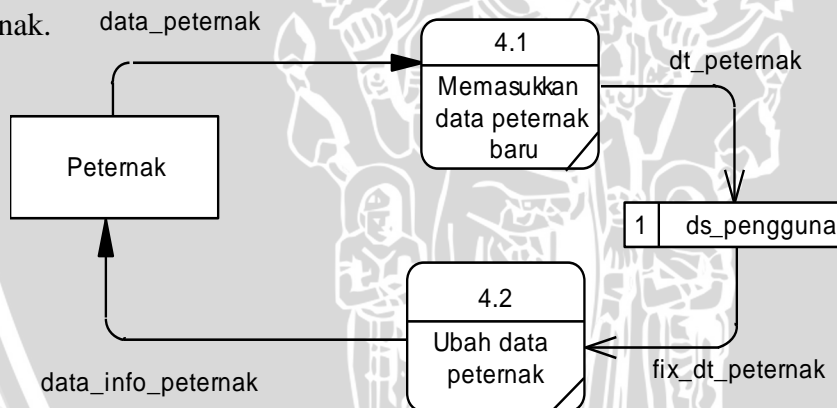
Gambar 4.15 DFD Level 2 Olah Data Sapi Bali  
 Sumber : [Perancangan]

Proses olah data pengguna didekomposisi menjadi beberapa subproses antara lain tambah data pengguna, ubah data pengguna dan hapus data pengguna. Pada subproses ini petugas BPTU dapat menambah pengguna sistem (peternak atau petugas BPTU), mengubah data pengguna, dan menghapus data pengguna. Gambar 4.16 menunjukkan DFD Level 2 dari proses olah data pengguna.



Gambar 4.16 DFD Level 2 Olah Data Pengguna  
 Sumber : [Perancangan]

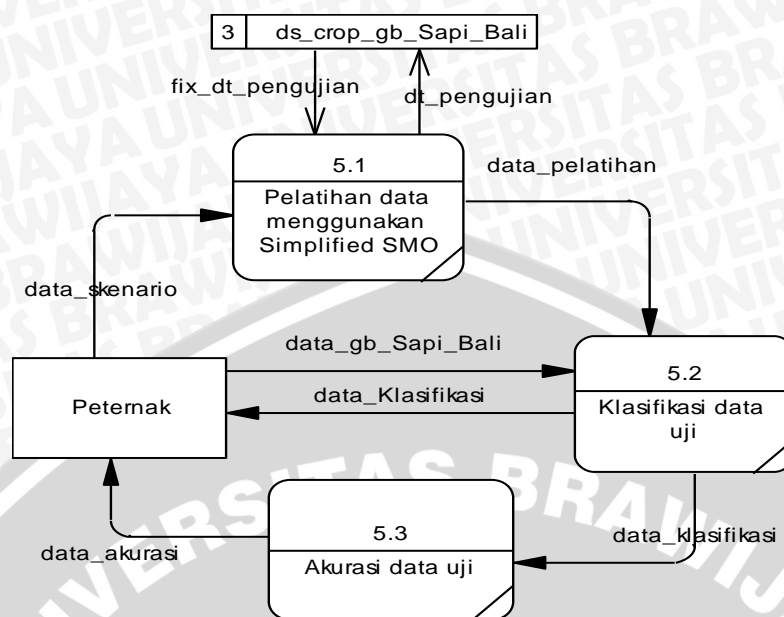
Proses olah data peternak mempunyai subproses ubah data pengguna. Pada subproses ini peternak dapat mengubah data peternak seperti nama, alamat, jenis kelamin, dan *password*. Gambar 4.17 menunjukkan DFD Level 2 dari proses olah data peternak.



Gambar 4.17 DFD Level 2 Olah Data Peternak  
 Sumber : [Perancangan]

Proses pengujian didekomposisi menjadi beberapa subproses antara lain pelatihan data menggunakan *Simplified SMO*, klasifikasi data uji dan akurasi data uji. Pada subproses pelatihan data menggunakan *Simplified SMO* peternak menginputkan nilai pada parameter yang dibutuhkan untuk melakukan poses pelatihan data. Hasil dari pelatihan data ini berupa nilai optimal  $\alpha$  dan *bias*. Gambar 4.18 menunjukkan DFD Level 2 dari proses pengujian.

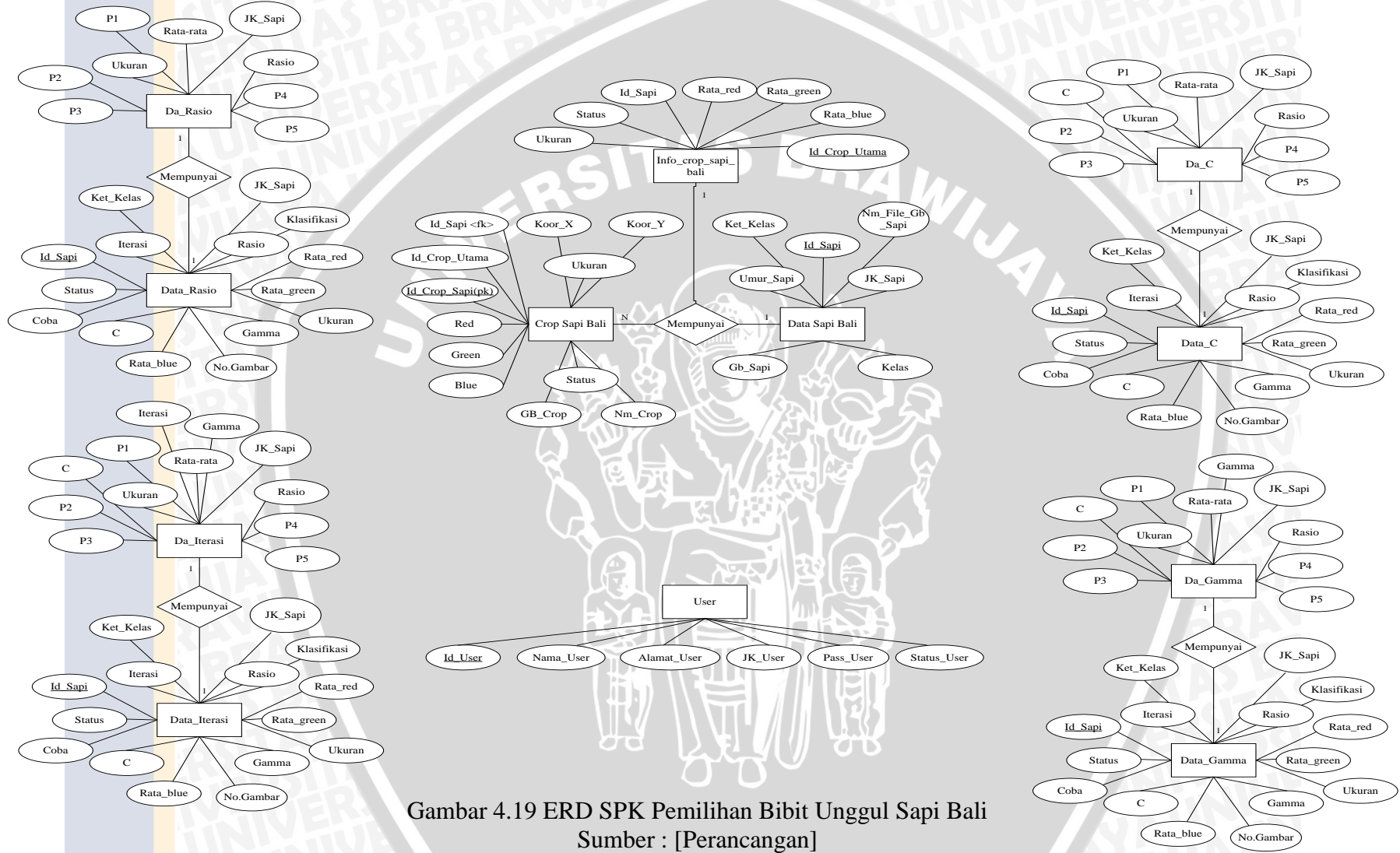




Gambar 4.18 DFD Level 2 Pengujian  
 Sumber : [Perancangan]

#### 4.2.2.2 Entity Relation Diagram (ERD)

ERD merupakan sebuah diagram yang dipakai untuk menjelaskan hubungan antar data dalam *database* berdasarkan entitas data yang mempunyai hubungan antar relasi. ERD sistem pendukung keputusan pemilihan bibit unggul Sapi Bali ini terdapat 12 entitas yaitu user, data Sapi Bali, crop Sapi Bali, info crop Sapi Bali, data rasio, data C, data gamma, data iterasi, da rasio, da C, da gamma dan da iterasi. Rancangan ERD pada sistem ini ditunjukkan pada gambar 4.19.

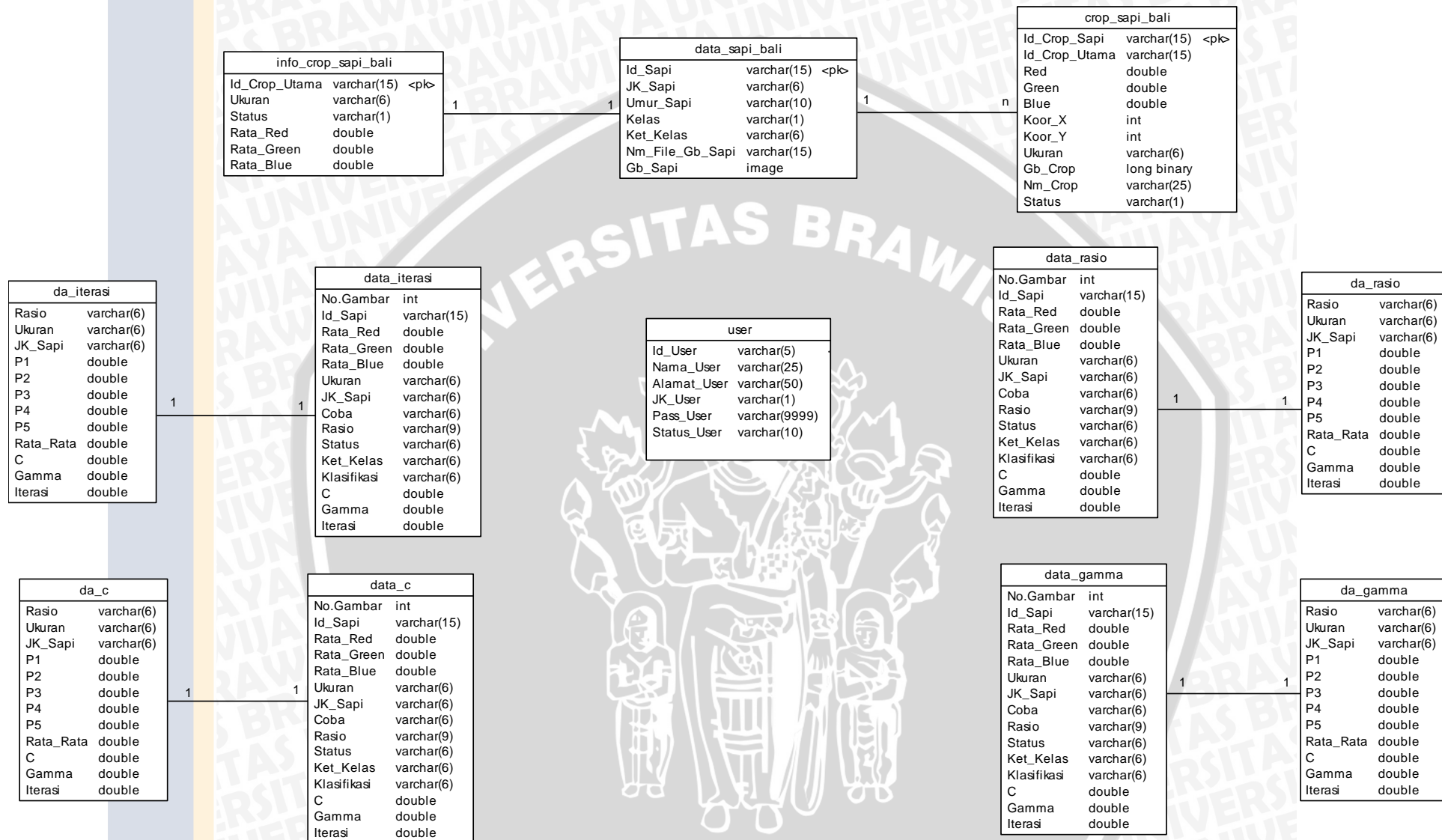


Gambar 4.19 ERD SPK Pemilihan Bibit Unggul Sapi Bali  
 Sumber : [Perancangan]

Entitas data Sapi Bali mempunyai hubungan dengan entitas crop data Sapi Bali dan info crop Sapi Bali. Pada entitas crop Sapi Bali berisi nilai-nilai fitur RGB, gambar *cropping* masing-masing ukuran dan titik koordinat *cropping* serta id Sapi yang berperan sebagai *foreign key* untuk entitas data Sapi Bali. Pada entitas info crop Sapi Bali berisi nilai rata-rata RGB dari tiga buah *cropping* serta id Sapi yang berperan sebagai *foreign key* pada entitas info crop Sapi Bali untuk entitas data Sapi Bali. Entitas data rasio, data C, data gamma dan data iterasi berfungsi untuk menyimpan data latih dan data uji yang dilakukan tiap skenario pengujian. Entitas da rasio, da C, da gamma dan da iterasi berfungsi untuk menyimpan hasil akurasi yang telah dilakukan untuk tiap skenario pengujian.

#### 4.2.2.3 Physical Data Model (PDM)

Sistem ini menggunakan *server Database Management System (DBMS)* yaitu MySQL. *Physical Data Model (PDM)* merupakan sebuah struktur fisik (desain nyata) dari *database* yang dibutuhkan dalam pembuatan sistem ini. Pada PDM ini terdapat 12 tabel yang akan digunakan antara lain user, data Sapi Bali, crop Sapi Bali, info crop Sapi Bali, data rasio, data C, data gamma, data iterasi, da rasio, da C, da gamma dan da iterasi. Gambar 4.20 menunjukkan Physical Data Model pada sistem ini.



Gambar 4.20 PDM SPK Pemilihan Bibit Unggul Sapi Bali  
 Sumber : [Perancangan]

Struktur dari masing-masing tabel pada *Physical Data Model* SPK pemilihan bibit unggul Sapi Bali sebagai berikut:

### 1. Tabel ‘User’

Tabel user digunakan untuk menyimpan informasi yang terdapat pada pengguna seperti id user, nama user, alamat user, jenis kelamin user, *password* user dan status user.

Tabel 4.4 Struktur Tabel User

No	Nama Field	Tipe	Lebar	Keterangan
1	Id_User	Varchar	5	Primary Key
2	Nama_User	Varchar	25	-
3	Alamat_User	Varchar	30	-
4	JK_User	Varchar	1	-
5	Pass_User	Varchar	9999	-
6	Status_User	Varchar	5	-

Sumber: [Perancangan]

### 2. Tabel ‘Data Sapi Bali’

Tabel data Sapi Bali untuk menyimpan data dan informasi dari citra Sapi Bali yang disimpan. Tabel ini menyimpan data id Sapi, JK Sapi, Umur Sapi, kelas, keterangan kelas, nama file gambar dan citra Sapi.

Tabel 4.5 Struktur Tabel Data Sapi Bali

No	Nama Field	Tipe	Lebar	Keterangan
1	Id_Sapi	Varchar	15	Primary Key
2	JK_Sapi	Varchar	6	-
3	Umur_Sapi	Varchar	10	-
4	Kelas	Varchar	1	-
5	Ket_Kelas	Varchar	6	-
6	Nm_File_Gb_Sapi	Varchar	15	-
7	Gb_Sapi	Image	-	-

Sumber: [Perancangan]

### 3. Tabel ‘Crop Sapi Bali’

Tabel crop Sapi Bali untuk menyimpan data dan informasi dari hasil cropping citra Sapi Bali yang disimpan. Tabel ini menyimpan data id crop Sapi, id crop utama, id Sapi, Red, Green, Blue, Koor X, Koor Y, Ukuran, Gb Crop, Nm Crop, Status.

Tabel 4.6 Struktur Tabel Crop Sapi Bali

No	Nama Field	Tipe	Lebar	Keterangan
1	Id_Crop_Sapi	Varchar	15	Primary Key
2	Id_Crop_Utama	Varchar	15	Foreign Key dari tabel Info crop Sapi Bali
3	Id_Sapi	Varchar	15	Foreign Key dari tabel data Sapi Bali

4	Red	Double	-	-
5	Green	Double	-	-
6	Blue	Double	-	-
7	Koor_X	Integer	-	-
8	Koor_Y	Integer	-	-
9	Ukuran	Varchar	6	-
10	Gb_Crop	Image	-	-
11	Nm_Crop	Varchar	15	-
12	Status	Varchar	1	-

Sumber: [Perancangan]

#### 4. Tabel 'Info Crop Sapi Bali'

Tabel info crop Sapi Bali untuk menyimpan data Sapi Bali. Tabel ini menyimpan data id crop utama, id Sapi, ukuran dan status.

Tabel 4.7 Struktur Tabel Info Crop Sapi Bali

No	Nama Field	Tipe	Lebar	Keterangan
1	Id_Crop_Utama	Varchar	15	Primary Key
2	Id_Sapi	Varchar	15	Foreign Key dari tabel data Sapi Bali
3	Ukuran	Varchar	6	-
4	Status	Varchar	1	-
5	Rata_red	Double	-	-
6	Rata_green	Double	-	-
7	Rata_blue	Double	-	-

Sumber: [Perancangan]

#### 5. Tabel 'Data Rasio'

Tabel data rasio untuk menyimpan data latih dan data uji pada setiap skenario pengujian rasio. Tabel ini menyimpan data no.gambar, id Sapi, rata red, rata green, rata blue, ukuran, jk sapi, coba, rasio, status, ket kelas, klasifikasi, C, gamma dan iterasi.

Tabel 4.8 Struktur Tabel Data Rasio

No	Nama Field	Tipe	Lebar	Keterangan
1	No. Gambar	Int	255	-
2	Id_Sapi	Varchar	15	-
3	Rata_red	Double	-	-
4	Rata_green	Double	-	-
5	Rata_blue	Double	-	-
6	Ukuran	Varchar	6	-
7	JK_Sapi	Varchar	6	-
8	Coba	Varchar	6	-
9	Rasio	Varchar	9	-
10	Status	Varchar	6	-
11	Ket_Kelas	Varchar	6	-
12	Klasifikasi	Varchar	6	-
13	C	Double	-	-
14	Gamma	Double	-	-

15	Iterasi	Double	-	-
----	---------	--------	---	---

Sumber: [Perancangan]

**6. Tabel ‘Data C’**

Tabel data C untuk menyimpan data latih dan data uji pada setiap skenario pengujian C. Tabel ini menyimpan data no.gambar, id Sapi, rata red, rata green, rata blue, ukuran, jk sapi, coba, rasio, status, ket kelas, klasifikasi, C, gamma dan iterasi.

Tabel 4.9 Struktur Tabel Data C

No	Nama Field	Tipe	Lebar	Keterangan
1	No. Gambar	Int	255	-
2	Id_Sapi	Varchar	15	-
3	Rata_red	Double	-	-
4	Rata_green	Double	-	-
5	Rata_blue	Double	-	-
6	Ukuran	Varchar	6	-
7	JK_Sapi	Varchar	6	-
8	Coba	Varchar	6	-
9	Rasio	Varchar	9	-
10	Status	Varchar	6	-
11	Ket_Kelas	Varchar	6	-
12	Klasifikasi	Varchar	6	-
13	C	Double	-	-
14	Gamma	Double	-	-
15	Iterasi	Double	-	-

Sumber: [Perancangan]

**7. Tabel ‘Data Gamma’**

Tabel data gamma untuk menyimpan data latih dan data uji pada setiap skenario pengujian gamma. Tabel ini menyimpan data no.gambar, id Sapi, rata red, rata green, rata blue, ukuran, jk sapi, coba, rasio, status, ket kelas, klasifikasi, C, gamma dan iterasi.

Tabel 4.10 Struktur Tabel Data Gamma

No	Nama Field	Tipe	Lebar	Keterangan
1	No. Gambar	Int	255	-
2	Id_Sapi	Varchar	15	-
3	Rata_red	Double	-	-
4	Rata_green	Double	-	-
5	Rata_blue	Double	-	-
6	Ukuran	Varchar	6	-
7	JK_Sapi	Varchar	6	-
8	Coba	Varchar	6	-
9	Rasio	Varchar	9	-
10	Status	Varchar	6	-
11	Ket_Kelas	Varchar	6	-



12	Klasifikasi	Varchar	6	-
13	C	Double	-	-
14	Gamma	Double	-	-
15	Iterasi	Double	-	-

Sumber: [Perancangan]

### 8. Tabel 'Data Iterasi'

Tabel data iterasi untuk menyimpan data latih dan data uji pada setiap skenario pengujian iterasi. Tabel ini menyimpan data no.gambar, id Sapi, rata red, rata green, rata blue, ukuran, jk sapi, coba, rasio, status, ket kelas, klasifikasi, C, gamma dan iterasi.

Tabel 4.11 Struktur Tabel Data Iterasi

No	Nama Field	Tipe	Lebar	Keterangan
1	No. Gambar	Int	255	-
2	Id_Sapi	Varchar	15	-
3	Rata_red	Double	-	-
4	Rata_green	Double	-	-
5	Rata_blue	Double	-	-
6	Ukuran	Varchar	6	-
7	JK_Sapi	Varchar	6	-
8	Coba	Varchar	6	-
9	Rasio	Varchar	9	-
10	Status	Varchar	6	-
11	Ket_Kelas	Varchar	6	-
12	Klasifikasi	Varchar	6	-
13	C	Double	-	-
14	Gamma	Double	-	-
15	Iterasi	Double	-	-

Sumber: [Perancangan]

### 9. Tabel 'Da Rasio'

Tabel da rasio untuk menyimpan hasil akurasi percobaan pada setiap skenario pengujian rasio. Tabel ini menyimpan data rasio, ukuran, jk sapi , p1, p2, p3, p4, p5 dan rata-rata.

Tabel 4.12 Struktur Tabel Da Rasio

No	Nama Field	Tipe	Lebar	Keterangan
1	Rasio	Varchar	6	-
2	Ukuran	Varchar	6	-
3	JK_Sapi	Varchar	6	-
4	P1	Double	-	-
5	P2	Double	-	-
6	P3	Double	-	-
7	P4	Double	-	-
8	P5	Double	-	-
9	Rata_rata	Double	-	-

Sumber: [Perancangan]



### 10. Tabel 'Da C'

Tabel da C untuk menyimpan hasil akurasi percobaan pada setiap skenario pengujian C. Tabel ini menyimpan data rasio, ukuran, jk sapi , p1, p2, p3, p4, p5, rata-rata dan C.

Tabel 4.13 Struktur Tabel Da C

No	Nama Field	Tipe	Lebar	Keterangan
1	Rasio	Varchar	6	-
2	Ukuran	Varchar	6	-
3	JK_Sapi	Varchar	6	-
4	P1	Double	-	-
5	P2	Double	-	-
6	P3	Double	-	-
7	P4	Double	-	-
8	P5	Double	-	-
9	Rata_rata	Double	-	-
10	C	Double	-	-

Sumber: [Perancangan]

### 11. Tabel 'Da Gamma'

Tabel da gamma untuk menyimpan hasil akurasi percobaan pada setiap skenario pengujian gamma. Tabel ini menyimpan data rasio, ukuran, jk sapi , p1, p2, p3, p4, p5, rata-rata, C dan gamma.

Tabel 4.14 Struktur Tabel Da Gamma

No	Nama Field	Tipe	Lebar	Keterangan
1	Rasio	Varchar	6	-
2	Ukuran	Varchar	6	-
3	JK_Sapi	Varchar	6	-
4	P1	Double	-	-
5	P2	Double	-	-
6	P3	Double	-	-
7	P4	Double	-	-
8	P5	Double	-	-
9	Rata_rata	Double	-	-
10	C	Double	-	-
11	Gamma	Double	-	-

Sumber: [Perancangan]

### 12. Tabel 'Da Iterasi'

Tabel da iterasi untuk menyimpan hasil akurasi percobaan pada setiap skenario pengujian iterasi. Tabel ini menyimpan data rasio, ukuran, jk sapi , p1, p2, p3, p4, p5, rata-rata, C, gamma dan iterasi.

Tabel 4.15 Struktur Tabel Da Iterasi

No	Nama Field	Tipe	Lebar	Keterangan
1	Rasio	Varchar	6	-

2	Ukuran	Varchar	6	-
3	JK_Sapi	Varchar	6	-
4	P1	Double	-	-
5	P2	Double	-	-
6	P3	Double	-	-
7	P4	Double	-	-
8	P5	Double	-	-
9	Rata_rata	Double	-	-
10	C	Double	-	-
11	Gamma	Double	-	-
12	Iterasi	Double	-	-

Sumber: [Perancangan]

### 4.2.3 Subsistem Manajemen Model

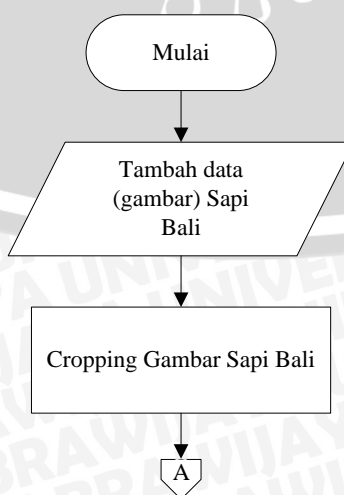
Subsistem manajemen model yang akan dilakukan meliputi pemodelan olah data Sapi Bali, pemodelan olah data pengguna, pemodelan algoritma *Simplified SMO* yang digunakan proses pelatihan dengan menggunakan fitur rata-rata RGB yang didapatkan dari ekstraksi fitur *cropping* gambar Sapi Bali dan pemodelan metode *one-against-all* dalam penentuan klasifikasi *multi-class*.

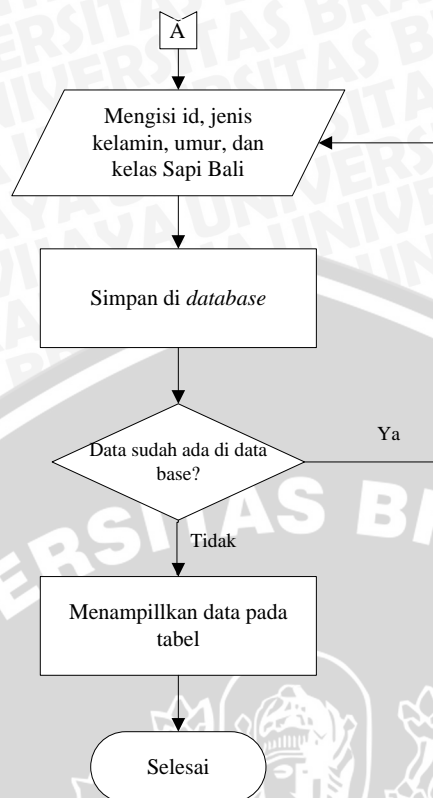
#### 4.2.4.1 Pemodelan Olah Data Sapi Bali

Proses olah data Sapi Bali dilakukan dengan menambah, mengubah dan menghapus data Sapi Bali. Data Sapi Bali ini akan digunakan untuk proses pengujian pada tahap selanjutnya.

##### 1. Proses penambahan data Sapi Bali

Proses penambahan data Sapi Bali ini berupa id Sapi, jenis kelamin Sapi, umur Sapi, kelas Sapi dan citra Sapi Bali yang diambil dari direktori file pada komputer. Diagram alir proses menambah data Sapi Bali dapat dilihat pada gambar 4.21.





Gambar 4.21 Diagram Alir Proses Tambah Data Sapi Bali  
Sumber: [Perancangan]

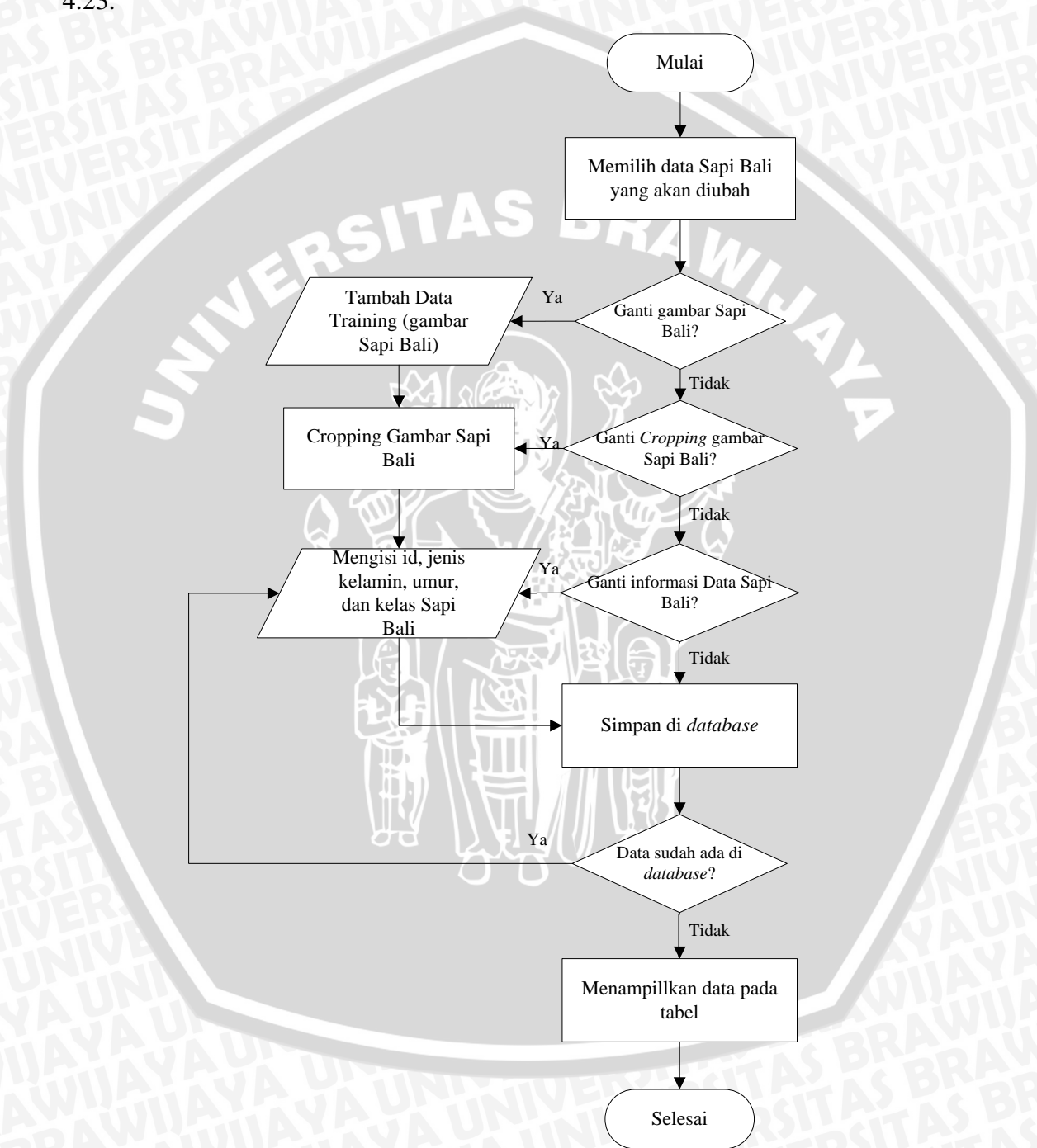
*Pseudocode* proses penambahan data Sapi Bali sebagai berikut :

<p><b>Nama Pseudocode</b> : menambah data Sapi Bali</p> <p><b>Deklarasi</b> :</p> <ul style="list-style-type: none"> <li>➤ String : id Sapi, jenis Kelamin Sapi, umur Sapi, kelas Sapi</li> <li>➤ Image : citra Sapi bali, cropp Sapi Bali</li> </ul> <p><b>Deskripsi</b> :</p> <ul style="list-style-type: none"> <li>➤ Input : citra, id, jenis kelamin, umur, kelas Sapi Bali</li> <li>➤ Proses :           <ol style="list-style-type: none"> <li>a) Memilih citra Sapi Bali yang akan ditambahkan ke dalam sistem.</li> <li>b) Melakukan proses <i>cropping</i> pada bagian samping dari citra tubuh Sapi Bali dengan ukuran tertentu.</li> <li>c) Mengisi id, jenis kelamin, umur dan kelas pada citra Sapi Bali ke dalam text field atau pilihan yang telah disediakan.</li> <li>d) Jika id Sapi Bali sudah ada di <i>database</i> maka akan muncul peringatan bahwa data sudah ada.</li> <li>e) Jika id Sapi Bali belum tersimpan di <i>database</i> maka data tersebut akan otomatis tersimpan dan akan ditambahkan pada tabel yang telah disediakan.</li> </ol> </li> </ul>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Gambar 4.22 *Pseudocode* Tambah Data Sapi Bali  
Sumber: [Perancangan]

2. Proses mengubah data Sapi Bali

Proses mengubah data Sapi Bali ini berupa id Sapi, jenis kelamin Sapi, umur Sapi, kelas Sapi dan citra Sapi Bali. Atribut id Sapi pada proses ini tidak dapat diubah. Diagram alir proses mengubah data Sapi Bali dapat dilihat pada gambar 4.23.



Gambar 4.23 Diagram Alir Proses Ubah Data Sapi Bali  
 Sumber: [Perancangan]

*Pseudocode* proses mengubah data Sapi Bali sebagai berikut :

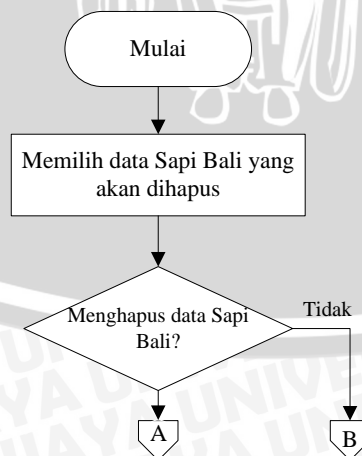
<p><b>Nama Pseudocode</b> : mengubah data Sapi Bali</p> <p><b>Deklarasi</b> :</p> <ul style="list-style-type: none"> <li>➤ String : id Sapi, jenis Kelamin Sapi, umur Sapi, kelas Sapi</li> <li>➤ Image : citra Sapi bali, cropp Sapi Bali</li> </ul> <p><b>Deskripsi</b> :</p> <ul style="list-style-type: none"> <li>➤ Input : citra, id, jenis kelamin, umur, kelas Sapi Bali</li> <li>➤ Proses:           <ol style="list-style-type: none"> <li>a) Memilih id Sapi Bali yang akan diubah pada tabel.</li> <li>b) Jika ingin mengganti citra Sapi Bali dapat melakukan proses memilih gambar Sapi Bali yang akan diganti pada direktori file di komputer.</li> <li>c) Jika ingin mengganti crop gambar Sapi Bali dapat melakukan proses cropping pada bagian samping dari gambar tubuh Sapi Bali.</li> <li>d) Jika ingin mengganti jenis kelamin, umur dan kelas dapat melakukan proses mengisi jenis kelamin, umur dan kelas pada citra Sapi Bali ke dalam text field atau pilihan yang telah disediakan.</li> <li>e) Jika telah selesai melakukan proses perubahan data Sapi Bali makan perubahan tersebut akan tersimpan otomatis di dalam database dan data pada tabel akan berubah sesuai dengan inputan.</li> </ol> </li> </ul>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

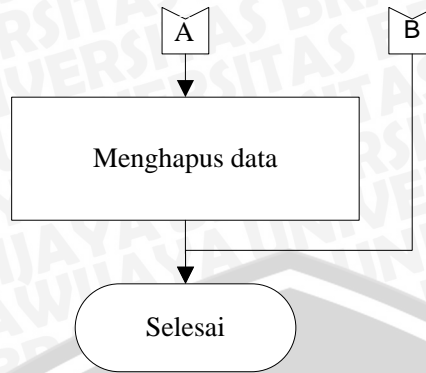
Gambar 4.24 *Pseudocode* Ubah Data Sapi Bali

Sumber: [Perancangan]

### 3. Proses menghapus data Sapi Bali

Proses menghapus data Sapi Bali ini merupakan suatu proses apabila terjadi kesalahan dalam penginputan data atau data tidak digunakan lagi. Diagram alir proses menghapus data Sapi Bali dapat dilihat pada gambar 4.25.





Gambar 4.25 Diagram Alir Proses Hapus Data Sapi Bali  
Sumber: [Perancangan]

*Pseudocode* proses menghapus data Sapi Bali sebagai berikut :

<p><b>Nama Pseudocode</b> : menghapus data Sapi Bali</p> <p><b>Deklarasi</b> :</p> <ul style="list-style-type: none"> <li>➤ String : id Sapi, jenis Kelamin Sapi, umur Sapi, kelas Sapi</li> <li>➤ Image : citra Sapi bali, cropp Sapi Bali</li> </ul> <p><b>Deskripsi</b> :</p> <ul style="list-style-type: none"> <li>➤ Input : id Sapi Bali</li> <li>➤ Proses :           <ol style="list-style-type: none"> <li>a) Memilih id Sapi Bali pada tabel yang akan di hapus.</li> <li>b) Data Sapi Bali pada tabel dan database akan otomatis terhapus jika dilakukan proses penghapusan data.</li> </ol> </li> </ul>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

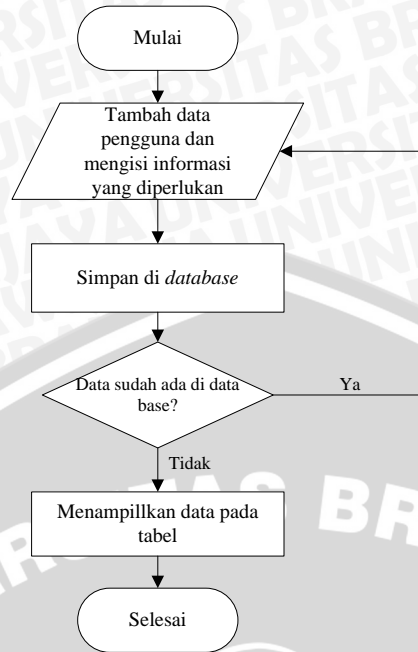
Gambar 4.26 *Pseudocode* Hapus Data Sapi Bali  
Sumber: [Perancangan]

#### 4.2.3.2 Pemodelan Olah Data Pengguna

Proses olah data pengguna dilakukan dengan menambah, mengubah dan menghapus data pengguna (peternak dan petugas BPTU) yang berinteraksi dengan sistem pendukung keputusan pemilihan bibit unggul Sapi Bali. Penambahan data pengguna hanya dilakukan oleh petugas BPTU.

##### 1. Proses penambahan data pengguna

Proses penambahan data pengguna ini berupa id pengguna, jenis kelamin pengguna, alamat pengguna, *password* pengguna dan status pengguna. Diagram alir proses menambah data pengguna dapat dilihat pada Gambar 4.27.



Gambar 4.27 Diagram Alir Proses Tambah Data Pengguna  
Sumber: [Perancangan]

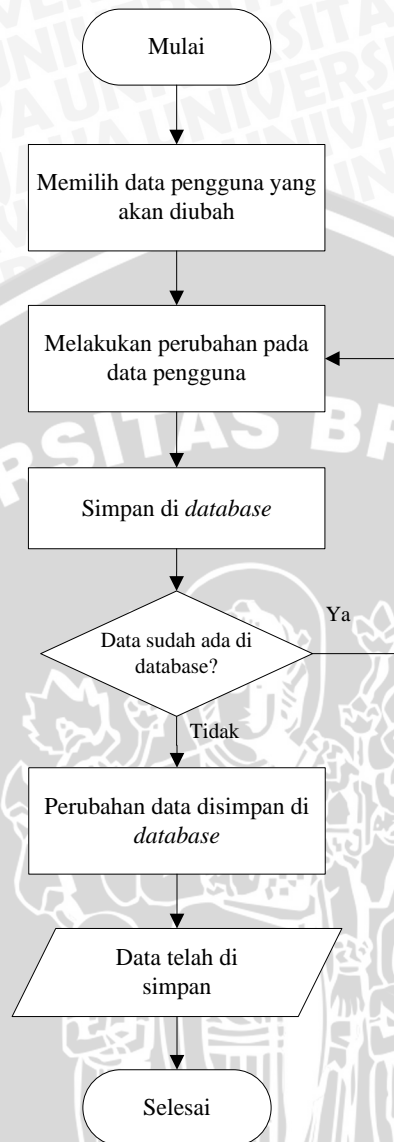
*Pseudocode* proses menambah data pengguna sebagai berikut :

<p><b>Nama Pseudocode</b> : menambah data pengguna</p> <p><b>Deklarasi</b> :</p> <ul style="list-style-type: none"> <li>➤ String : id pengguna, jenis kelamin pengguna, alamat pengguna, password pengguna, status pengguna</li> </ul> <p><b>Deskripsi</b> :</p> <ul style="list-style-type: none"> <li>➤ Input : id pengguna, jenis kelamin pengguna, alamat pengguna, password pengguna, status pengguna</li> <li>➤ Proses :           <ol style="list-style-type: none"> <li>a) Menambah data pengguna yang akan terlibat di dalam sistem.</li> <li>b) Memasukan informasi data pengguna berupa id, nama, jenis kelamin, alamat, password dan status pengguna yang dilakukan oleh petugas BPTU</li> <li>c) Data pengguna pada tabel dan database akan otomatis ditambahkan jika dilakukan proses pengambanan data pengguna.</li> <li>d) Jika ada data pengguna yang sama atau kesalahan dalam penambahan data sistem akan memberikan peringatan kepada petugas BPTU.</li> </ol> </li> </ul>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Gambar 4.28 *Pseudocode* Tambah Data Pengguna  
Sumber: [Perancangan]

## 2. Proses pengubahan data pengguna

Proses pengubahan data pengguna ini berupa id pengguna, jenis kelamin pengguna, alamat pengguna, *password* pengguna dan status pengguna. Diagram alir proses mengubah data pengguna dapat dilihat pada gambar 4.29.



Gambar 4.29 Diagram Alir Proses Ubah Data Pengguna  
Sumber: [Perancangan]

*Pseudocode* proses mengubah data pengguna sebagai berikut :

<p><b>Nama Pseudocode</b> : mengubah data pengguna</p> <p><b>Deklarasi</b> :</p> <ul style="list-style-type: none"> <li>➤ String : id pengguna, jenis kelamin pengguna, alamat user, password pengguna, status pengguna</li> </ul> <p><b>Deskripsi</b> :</p> <ul style="list-style-type: none"> <li>➤ Input : id pengguna, jenis kelamin pengguna, alamat pengguna, password pengguna, status pengguna</li> <li>➤ Proses :           <ol style="list-style-type: none"> <li>a) Memilih data pengguna yang akan diubah.</li> <li>b) Mengubah informasi pada data pengguna sesuai</li> </ol> </li> </ul>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

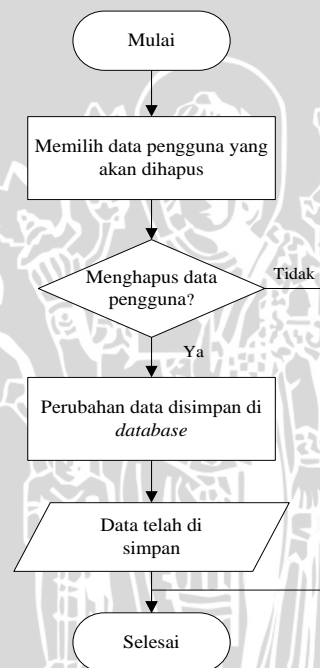


- dengan kebutuhan.
- c) Data pengguna pada tabel dan database akan otomatis diubah, jika dilakukan proses perubahan data pengguna.
  - d) Jika ada data pengguna yang sama atau kesalahan dalam perubahan data sistem akan memberikan peringatan kepada admin.

Gambar 4.30 Pseudocode Ubah Data Pengguna  
Sumber: [Perancangan]

### 3. Proses penghapusan data pengguna

Proses penghapusan data pengguna ini berupa id pengguna, jenis kelamin pengguna, alamat pengguna, *password* pengguna dan status pengguna. Diagram alir proses menghapus data pengguna dapat dilihat pada gambar 4.31.



Gambar 4.31 Diagram Alir Proses Hapus Data Pengguna  
Sumber: [Perancangan]

*Pseudocode* proses menghapus data user sebagai berikut :

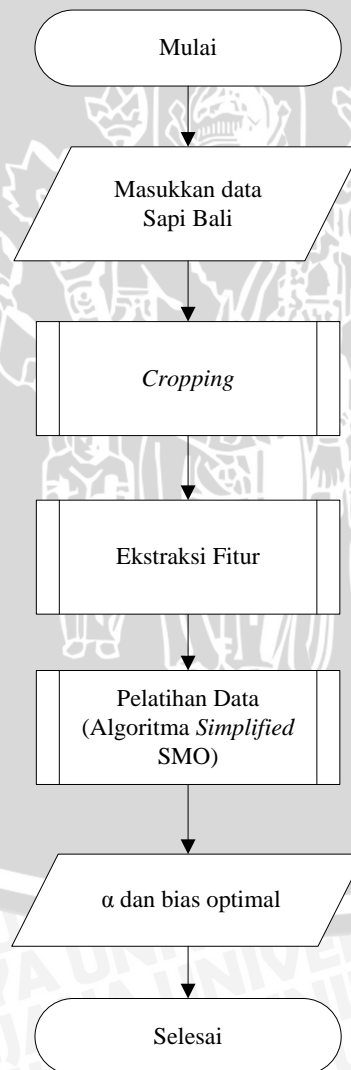
<b>Nama Pseudocode</b>	: menghapus data pengguna
<b>Deklarasi</b>	: ➤ String : id pengguna, jenis kelamin pengguna, alamat pengguna, password pengguna, status pengguna
<b>Deskripsi</b>	: ➤ Input : id pengguna, jenis kelamin pengguna, alamat pengguna, password pengguna, status pengguna ➤ Proses : a) Memilih data pengguna yang akan dihapus.

- b) Menghapus data pengguna sesuai pilihan.
- c) Menampilkan peringatan data berhasil dihapus

Gambar 4.32 *Pseudocode* Hapus Data Pengguna  
Sumber: [Perancangan]

#### 4.2.3.3 Pemodelan Algoritma *Simplified* SMO

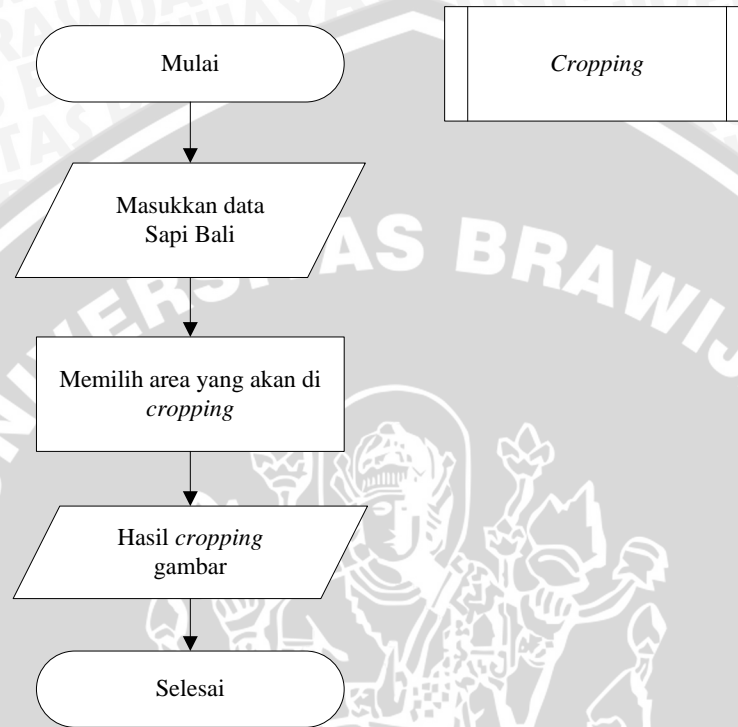
Pada pemodelan algoritma *Simplified* SMO pada penelitian ini digunakan untuk mendapatkan nilai optimal dari  $\alpha$  dan *bias* pada proses pelatihan data. Tahap awal penggunaan algoritma *Simplified* SMO terlebih dahulu dilakukan proses ekstraksi fitur pada citra *cropping* Sapi Bali untuk mendapatkan fitur rata-rata *red*, *green*, dan *blue* (RGB). Fitur tersebut akan digunakan dalam proses pelatihan dengan menggunakan algoritma *Simplified* SMO. Diagram alir penggunaan algoritma *Simplified* SMO secara umum dapat dilihat pada gambar 4.33.



Gambar 4.33 Diagram Alir Algoritma *Simplified* SMO  
Sumber: [Perancangan]

1. Proses *cropping* citra Sapi Bali

Proses *cropping* ini dilakukan pada citra Sapi Bali dengan cara memotong bagian (kulit) samping dari tubuh citra Sapi Bali berukuran tertentu. Diagram alir proses *cropping* citra Sapi Bali dapat dilihat pada gambar 4.34.



Gambar 4.34 Diagram Alir Proses *Cropping*  
 Sumber: [Perancangan]

*Pseudocode* proses *cropping* citra Sapi Bali sebagai berikut :

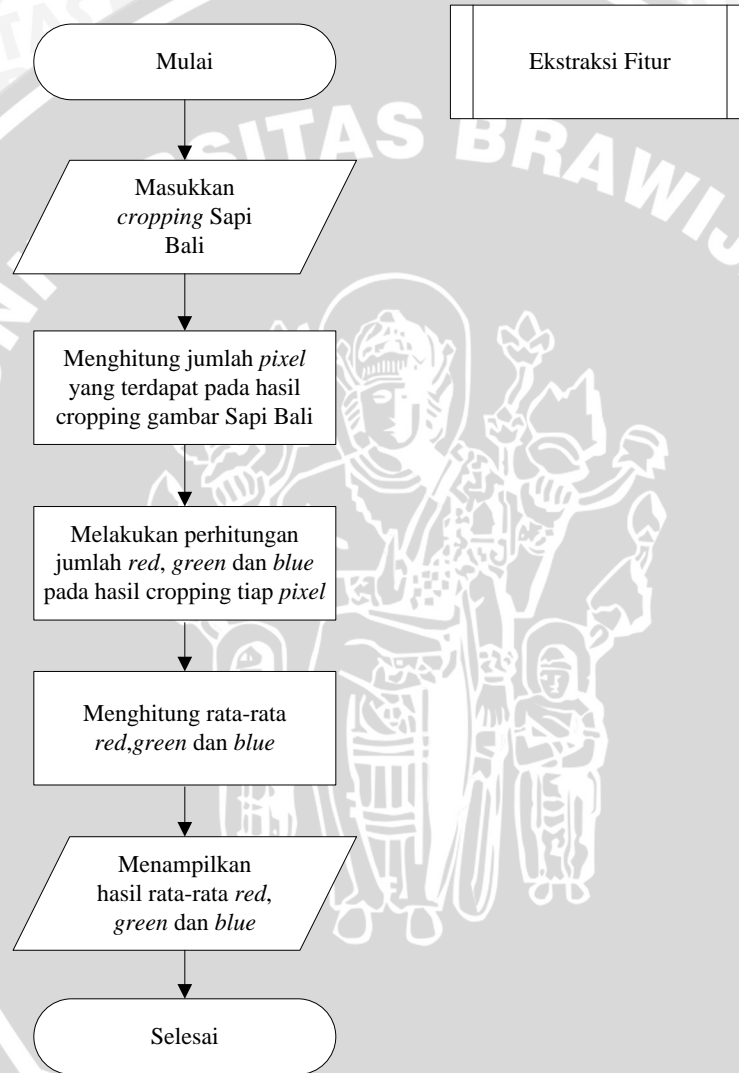
```

Nama Pseudocode : cropping citra Sapi Bali
Deklarasi :
    ➤ Image : citra Sapi Bali
Deskripsi :
    ➤ Input : citra Sapi Bali
    ➤ Proses :
        a) Citra Sapi Bali di masukkan ke dalam sistem
        b) Memilih area pada citra Sapi Bali yang akan di
           cropping sesuai dengan kebutuhan.
        c) Sistem akan menampilkan hasil cropping yang telah
           dilakukan dengan tiga buah ukuran cropping yang
           berbeda (128x128 pixel, 64x64 pixel dan 32x32
           pixel).
    
```

Gambar 4.35 *Pseudocode* *Cropping* Citra Sapi Bali  
 Sumber: [Perancangan]

2. Proses ekstraksi fitur

Proses ini dilakukan untuk mendapatkan fitur yang dibutuhkan dalam pelatihan menggunakan algoritma *Simplified SMO*. Fitur yang akan didapatkan dari proses ini adalah rata-rata *red*, *green* dan *blue* (RGB) dari *cropping* citra Sapi Bali yang telah dilakukan pada proses sebelumnya. Diagram alir untuk proses *cropping* dapat dilihat pada gambar 4.36.



Gambar 4.36 Diagram Alir Ekstraksi Fitur  
 Sumber: [Perancangan]

*Pseudocode* proses ekstraksi fitur sebagai berikut :

<p><b>Nama Pseudocode</b> : ekstraksi fitur</p> <p><b>Deklarasi</b> :</p> <ul style="list-style-type: none"> <li>➤ Image : <i>cropping</i> citra Sapi Bali</li> <li>➤ Int : <i>i, j, count, rtot, gtot, btot</i></li> </ul>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

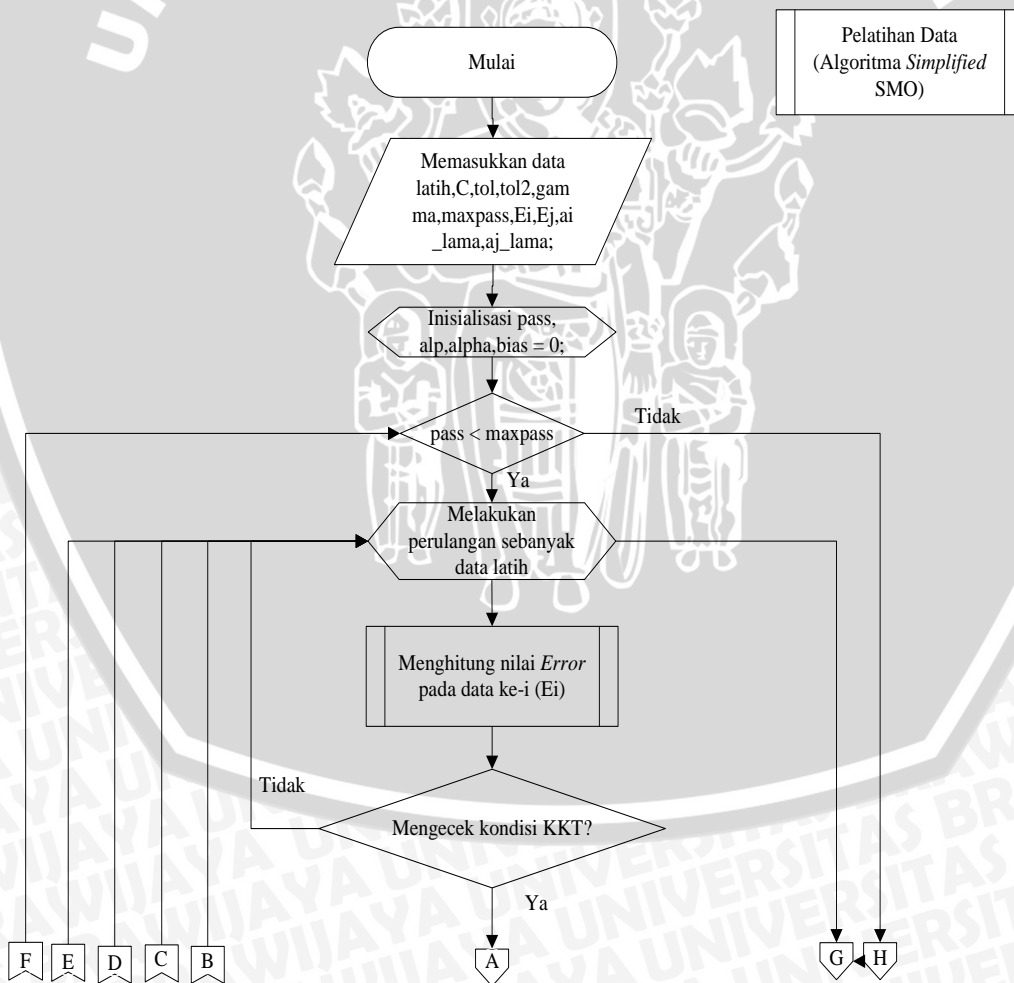
➤ Double : rataRed, rataGreen, rataBlue  
**Deskripsi :**  
 ➤ Input : *cropping* citra Sapi Bali  
 ➤ Proses :  
 a) *Cropping* citra Sapi Bali di masukkan ke dalam sistem.  
 b) Melakukan perhitungan jumlah nilai fitur red, green dan blue dengan ketentuan  $i = 0$  hingga  $<$  tinggi *cropping* dan  $j = 0$  hingga  $<$  lebar *cropping*  
 c) Menghitung rata-rata fitur dengan menjumlahkan total fitur dan membagi dengan jumlah totPixel.

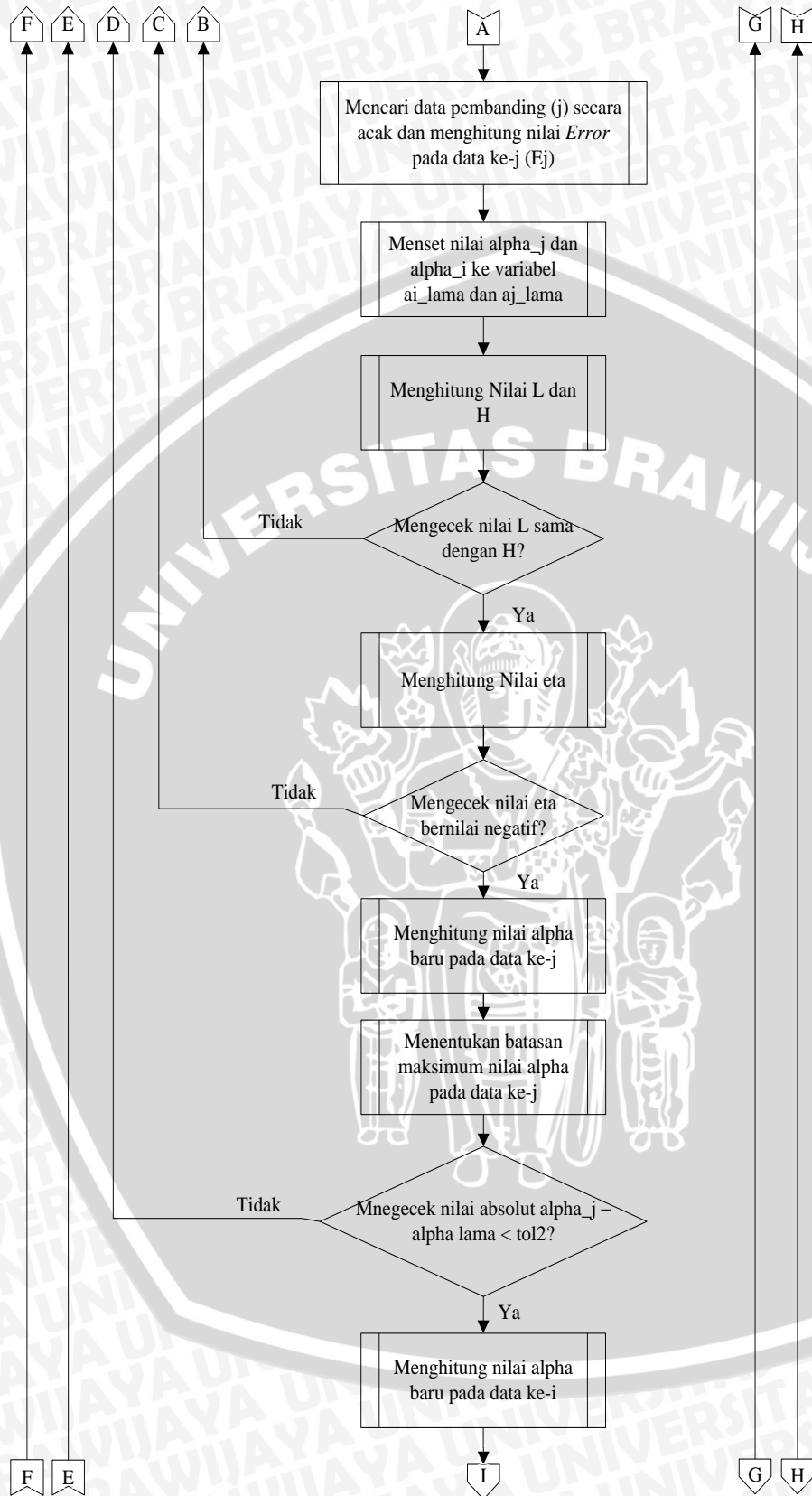
Gambar 4.37 Pseudoceode Ekstraksi Fitur

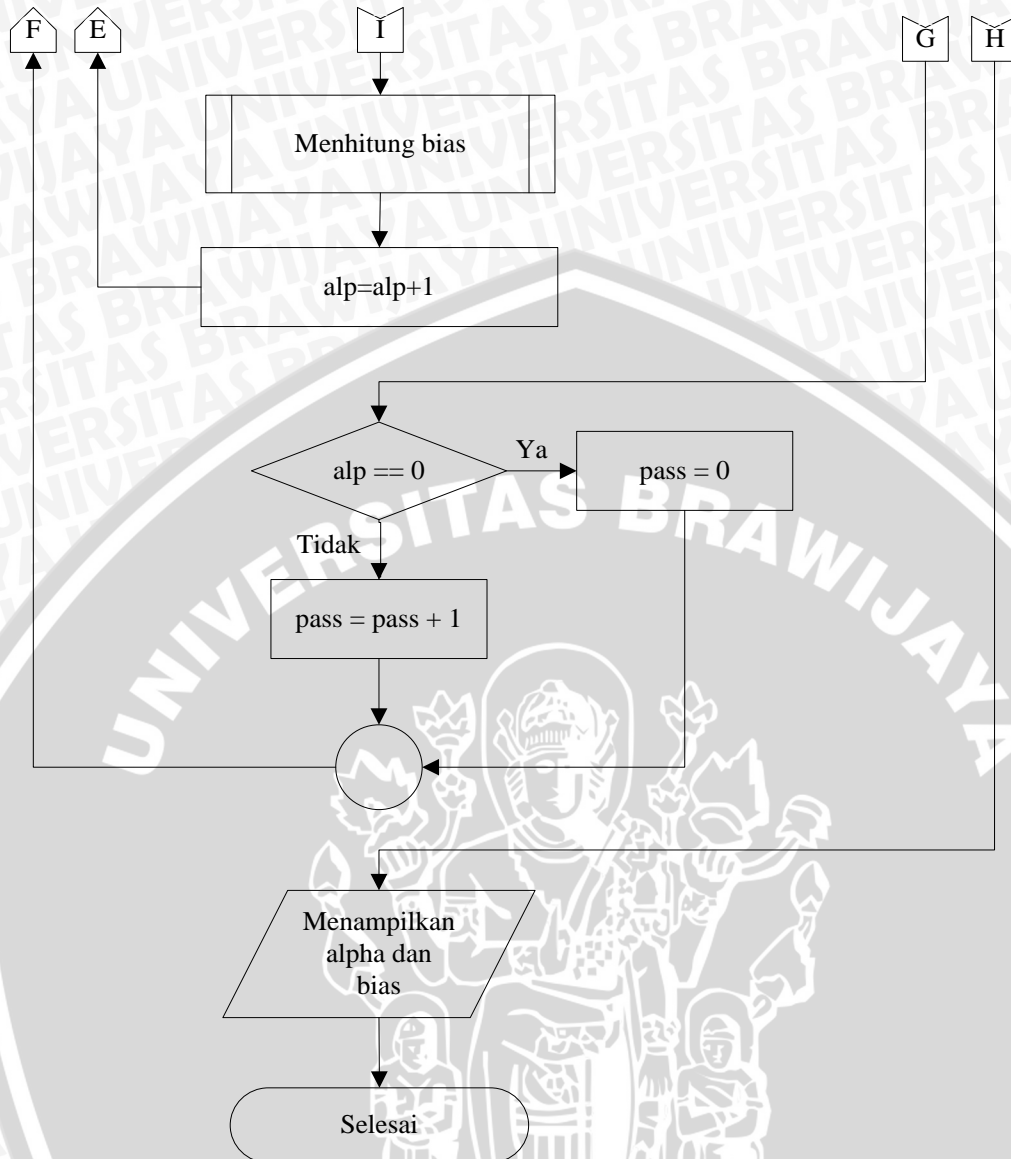
Sumber: [Perancangan]

3. Proses pelatihan menggunakan algoritma *Simplified SMO*

Proses ini dilakukan untuk mendapatkan nilai optimal  $\alpha$  dan *bias* berdasarkan fitur yang telah didapatkan dari proses ekstraksi fitur. Berikut diagram alir algoritma *Simplified SMO* dapat dilihat pada gambar 4.38.







Gambar 4.38 Diagram Alir Algoritma *Simplified SMO*  
 Sumber: [Perancangan]

*Pseudocode* proses algoritma *Simplified SMO* sebagai berikut :

<b>Nama Pseudocode</b>	: Pseudocode <i>Simplified SMO</i>
<b>Deklarasi</b>	:
➤ Int	: maxPass, pass
➤ Double	: C, tol, tol2, gamma, Ei, Ej, ai_lama, aj_lama, L, H, eta bias
<b>Deskripsi</b>	:
➤ Input	: gamma, C, data latih(fitur), maxpass
➤ Proses	:
	a) Memasukkan data latih(fitur), nilai gamma, C, maxpass.
	b) Menghitung nilai <i>error</i> pada data ke-i
	c) Mengecek kondisi KKT terhadap data ke-i, jika data ke-i tidak memenuhi kondisi KKT maka dilakukan

- perhitungan error pada data ke- $i$  ( $i++$ )
- d) Memilih data ke- $j$  sebagai data pembanding secara acak (*random*)
  - e) Menghitung nilai *error* pada data ke- $j$
  - f) Menset nilai  $\alpha_i$  dan  $\alpha_j$  awal ke dalam variabel  $\alpha_{i\_lama}$  dan  $\alpha_{j\_lama}$
  - g) Menghitung nilai L dan H, jika nilai L dan H sama maka perhitungan selanjutnya berhenti dan kembali ke perhitungan awal dengan menghitung error ke- $i$  ( $i++$ )
  - h) Menghitung nilai *eta*, jika nilai *eta* positif maka perhitungan selanjutnya berhenti dan kembali ke perhitungan awal dengan menghitung error ke- $i$  ( $i++$ )
  - i) Menghitung nilai alpha baru pada data ke- $j$  ( $\alpha_j$ )
  - j) Mengecek nilai alpha baru pada data ke- $j$  lebih besar dari H atau lebih kecil dari L
  - k) Mengecek nilai absolut dari  $|\alpha_j - \alpha_{j\_lama}| > tol2$ , jika memenuhi kondisi tersebut maka perhitungan selanjutnya berhenti dan kembali ke perhitungan awal dengan menghitung error ke- $i$  ( $i++$ )
  - l) Menghitung nilai alpha baru pada data ke- $i$  ( $\alpha_i$ )
  - m) Menghitung nilai *bias*
  - n) Menambah variabel  $alp = alp + 1$
  - o) Jika  $alp == 0$  maka  $pass++$ , jika  $alp != 0$  maka  $pass = 0$
  - p) Jika  $pass > maxpass$  maka perhitungan berhenti dan menampilkan nilai optimal  $\alpha$  dan *bias*

Gambar 4.39 Pseudocode Algoritma *Simplified SMO*

Sumber: [Perancangan]

Contoh perhitungan menggunakan algoritma *Simplified SMO* dalam proses pelatihan untuk mendapatkan nilai optimal dari  $\alpha$  dan *bias* pada kasus pemilihan bibit unggul Sapi Bali. Data latih yang digunakan dalam perhitungan ini menggunakan data Sapi Bali jantan dan ukuran *cropping*  $32 \times 32$  pixel sebanyak 6 data yang terbagi kedalam tiga kelas (Baik, Sedang dan Buruk).

Pada proses perhitungan dibutuhkan fitur rata-rata *red*, *green* dan *blue* (RGB) untuk melakukan proses pelatihan. Berikut proses perhitungan fitur rata-rata *red* (R) menggunakan persamaan (2-43) dengan mengacu representasi *cropping* citra Sapi Bali pada lampiran 5.

$$\begin{aligned}\bar{X}_R &= \frac{X_{11} + X_{12} + X_{13} + X_{21} + \dots + X_{3232}}{1024} \\ &= \frac{62 + 62 + 62 + 63 + \dots + 65}{1024} \\ &= \frac{71680}{1024}\end{aligned}$$



= 70

Proses perhitungan fitur rata-rata *green* (G) dan *blue* (B) sama seperti proses perhitungan fitur rata-rata *red* (R).

Tabel 4.16 Data Pelatihan Pertama

Data Citra ke-	Id Citra Sapi Bali	Fitur			Kelas	Keterangan Kelas	Target (y)
		R	G	B			
X1	SB-2	70	67	55	1	Baik	1
X2	SB-5	57	61	68	1	Baik	1
X3	SB-17	133	88	50	2	Sedang	1
X4	SB-20	164	124	96	2	Sedang	1
X5	SB-40	170	130	86	3	Buruk	-1
X6	SB-42	173	156	125	3	Buruk	-1

Sumber: [Perancangan]

Pada tabel 4.16 terdapat masing-masing dua data citra Sapi Bali untuk tiap kelas (Baik, Sedang dan Buruk). Pada pelatihan pertama data citra untuk kelas Baik dan Sedang memiliki target (y) +1 sedangkan data citra untuk kelas Buruk memiliki target (y) -1. Pelatihan pertama ini akan menghasilkan fungsi keputusan berdasarkan nilai optimal dari  $\alpha$  dan *bias* pada masing-masing data latih. Berikut adalah proses pelatihan pertama menggunakan algoritma *Simplified SMO*.

#### **Proses Pelatihan Data Latih Pertama (Baik vs Buruk)**

Pada proses pelatihan menggunakan algoritma *Simplified SMO* terlebih dahulu diberikan nilai parameter-parameter yang dibutuhkan yaitu  $\alpha = 0$ ,  $\text{bias\_awal} (b) = 0$ ,  $\text{tol} = 0,01$ ,  $\text{tol}2 = 0,0001$ ,  $C = 1$ ,  $\text{gamma} = 0,001$ , dan  $\text{maxIter} = 1$ . Nilai parameter C, gamma dan maxIter merupakan parameter bebas yang akan digunakan dalam proses pengujian.

Tahap awal proses pelatihan mencari nilai *dot product* dari masing-masing data latih menggunakan kernel *Radial Basis Function* (RBF) yang ditunjukkan oleh persamaan (2-17). Sebagai contoh nilai kernel pada data X1 dan X2 ( $K(x_1, x_2)$ ) dengan  $\text{gamma} = 0,001$  adalah

$$\begin{aligned}
 K(x_1, x_2) &= \exp\left(-\gamma \|x_i - x_j\|^2\right) \\
 &= \exp\left[-0,001 * ((70 - 57)^2 + (67 - 61)^2 + (55 - 68)^2)\right] \\
 &= \exp(-0,374) \\
 &= \mathbf{0,687976912}
 \end{aligned}$$

Berdasarkan perhitungan kernel diatas maka didapatkan nilai *dot product* untuk data latih seperti yang ditunjukkan oleh tabel 4.17.

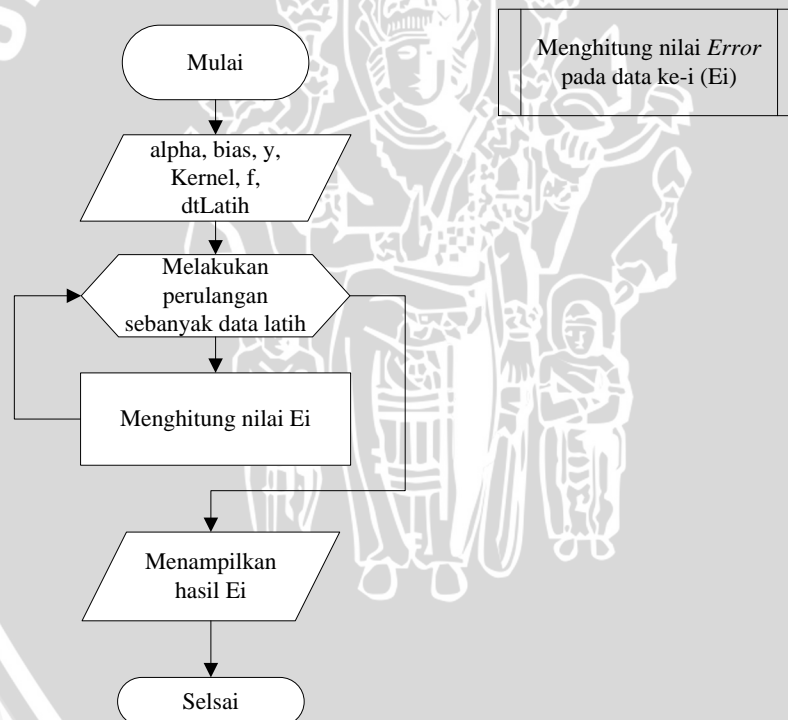
Tabel 4.17 Hasil Perhitungan dengan Kernel RBF

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

Sumber: [Perancangan]

- Tahap ke – 1

Mencari nilai *error* pada setiap data training (data ke-i) dengan menggunakan persamaan (2-36). Diagram alir untuk mencari nilai *error* dapat dilihat pada gambar 4.40.



Gambar 4.40 Diagram Alir Perhitungan *Error* Pada Data ke-i  
Sumber: [Perancangan]

*Pseudocode* proses perhitungan *error* data ke-i sebagai berikut :

<p><b>Nama Pseudocode</b> : perhitungan <i>error</i> data ke-i</p> <p><b>Deklarasi</b> :</p> <ul style="list-style-type: none"> <li>➤ double : f(x), E[i]</li> </ul> <p><b>Deskripsi</b> :</p> <ul style="list-style-type: none"> <li>➤ Input : alpha[i], bias, target(y[i]), kernel(x,i)</li> <li>➤ Proses:</li> </ul>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- Mengambil nilai dari  $\alpha$ ,  $\text{target}(y)$ ,  $b$  dan kernel pada tiap data latih
- Menghitung nilai  $\text{error}(E_i)$  menggunakan rumus yang telah ditentukan
- Menampilkan hasil  $E_i$

Gambar 4.41 Pseudocode Perhitungan Error Pada Data ke-i  
Sumber: [Perancangan]

Contoh perhitungan nilai  $\text{error}(E_i)$  pada data latih adalah misal mencari  $\text{error}(i)$  pada perulangan ke-1 dengan data ke-1,  $\alpha = 0$ ,  $b = 0$ ,  $\text{target}(y) = 1$

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

$$E_i = f(x) - y_i$$

$$f(x) = ((0 * 1 * 1) + (0 * 1 * 0,687976912) + (0 * 1 * 0,011855066) + (0 * 1 * 1,05076E - 06) + (0 * -1 * 3,28083E - 07) + (0 * -1 * 6,67545E - 11) + 0 = 0$$

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

Berdasarkan perhitungan  $\text{error}(i)$  di atas maka didapatkan nilai  $\text{error}$  pada tiap data training untuk masing-masing perulangan seperti yang ditunjukkan pada tabel 4.18.

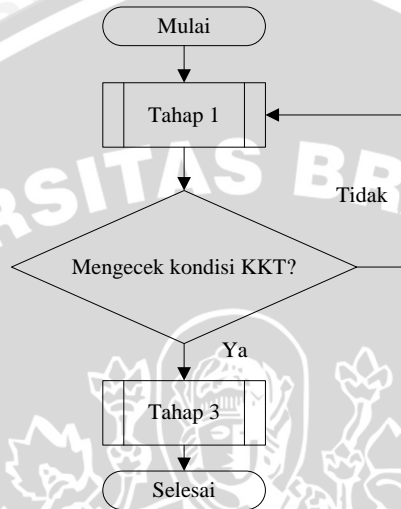
Tabel 4.18 Hasil Error pada data ke-i

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

Sumber: [Perancangan]

- Tahap ke – 2

Melakukan pengecekan terhadap kondisi *Karush-Kuhn-Tucker* (KKT) pada data training dan nilai  $E_i$ , jika tidak memenuhi kondisi tersebut maka perulangan kembali pada tahap ke – 1 dengan melakukan perhitungan error pada data ke-i selanjutnya. Diagram alir untuk pengecekan terhadap kondisi KKT dapat dilihat pada gambar 4.42.



Gambar 4.42 Diagram Alir Pengecekan kondisi KKT  
Sumber: [Perancangan]

*Pseudocode* proses pengecekan kondisi KKT sebagai berikut :

<b>Nama Pseudocode</b> :	pengecekan kondisi KKT
<b>Deklarasi</b> :	
➤ double	: E[i], tol, tol2, C
<b>Deskripsi</b> :	
➤ Input	: alpha[i], target(y[i]), Ei, C
➤ Proses:	
	a) Memasukan nilai E[i] yang didapatkan dari tahap 1
	b) Melakukan pengecekan terhadap kondisi KKT.
	c) Jika memenuhi kondisi KKT maka dapat melanjutkan ke tahap ke 3, jika tidak memenuhi kembali ke tahap 1 dengan i++

Gambar 4.43 *Pseudocode* Pengecekan Kondisi KKT  
Sumber: [Perancangan]

Hasil dari kondisi KKT pada tiap data latih untuk masing-masing perulangan dapat dilihat pada tabel 4.19.

Tabel 4.19 Kondisi KKT

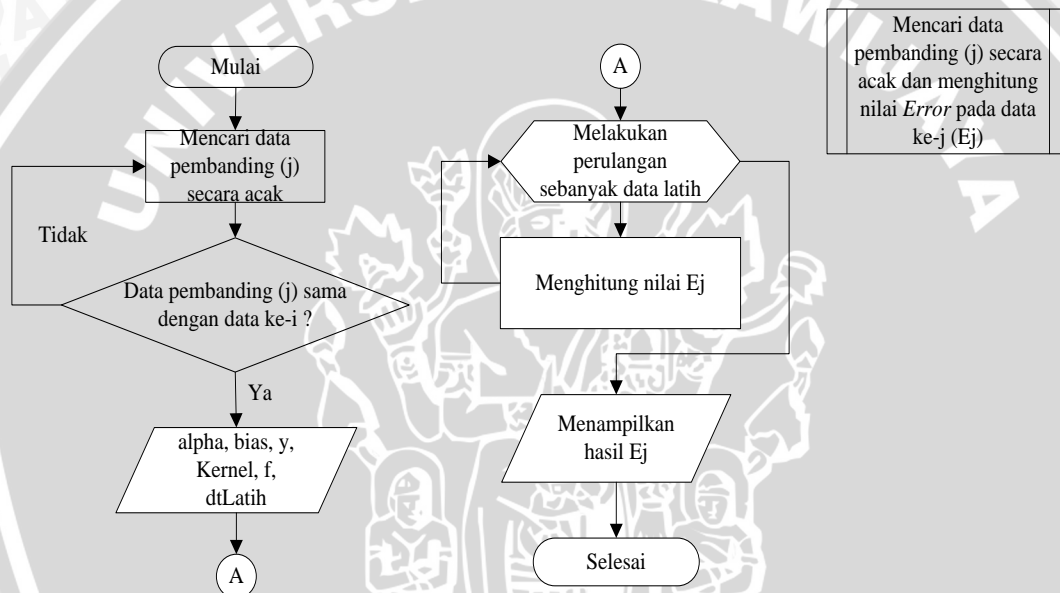
Perulangan ke-	Hasil	Perulangan ke-	Hasil
1	True	7	True

2	True	8	True
3	True	9	True
4	True	10	False
5	False	11	False
6	False	12	False

Sumber:[Perancangan]

- Tahap ke – 3

Tahap berikutnya adalah mencari data pembanding dari data latih secara *random* (acak) dengan ketentuan  $j \neq i$  dan mencari nilai *error* ( $E_j$ ) pada data tersebut. Diagram alir untuk mencari data pembanding secara *random* dapat dilihat pada gambar 4.44.



Gambar 4.44 Diagram Alir Mencari Data Pembanding dan Nilai *Error* Pada Data ke-j

Sumber: [Perancangan]

*Pseudocode* proses pencarian data pembanding dan nilai error sebagai berikut :

```

Nama Pseudoceode : pencarian data pembanding dan nilai error
Deklarasi :
    ➤ Int : j
    ➤ Double : E[j]
Deskripsi :
    ➤ Input : j (random)
    ➤ Proses:
        a) Mendapatkan nilai j secara random yang digunakan untuk menentukan data pembanding dari data latih yang tersedia.
        b) Melakukan perhitungan error E[j] menggunakan persamaan yang sama dengan menghitung nilai error
    
```



pada data ke- $i$ .  
c) Menampilkan hasil  $E_j$

Gambar 4.45 *Pseudocode* Mencari Data Pemanding dan Nilai *Error* Pada ke- $j$   
Sumber: [Perancangan]

Proses perhitungan nilai *error* pada data ke- $j$  sama seperti perhitungan nilai *error* pada data ke- $i$  menggunakan persamaan (2-36), tetapi untuk menentukan data pemanding dilakukan secara acak. Berikut contoh perhitungan nilai *error*( $j$ ) pada perulangan ke-1, misalkan nilai  $j = 3$ .

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

$$E_j = f(x) - y_j$$

$$f(x) = ((0 * 1 * 0,011855066) + (0 * 1 * 0,00108194) + (0 * 1 * 1) \\ + (0 * 1 * 0,012613344) + (0 * -1 * 0,01192641) \\ + (0 * -1 * 7,1457E - 06) + 0 = 0$$

$$E_3 = 0 - 1 = -1$$

Berdasarkan perhitungan *error* ( $E_j$ ) di atas maka didapatkan nilai *error* pada tiap data latih untuk masing-masing perulangan seperti yang ditunjukkan pada tabel 4.20.

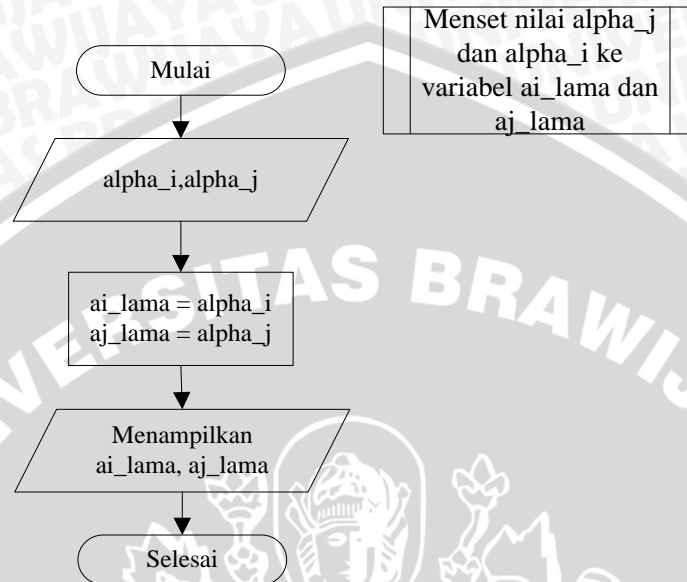
Tabel 4.20 *Error* data pemanding ( $E_j$ )

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

Sumber:[Perancangan]

- Tahap ke – 4

Men-*set* nilai alpha pada data *i* dan data *j* ke dalam variabel alpha\_lama. Diagram alir untuk menset nilai alpha ke dalam variabel alpha\_lama dapat dilihat pada gambar 4.46.



Gambar 4.46 Diagram Alir Men-*set* Nilai Alpha  
Sumber: [Perancangan]

*Pseudocode* proses men-*set* nilai alpha sebagai berikut :

<b>Nama Pseudocode</b> :	men- <i>set</i> nilai alpha
<b>Deklarasi</b> :	
➤ Double	: ai_lama, aj_lama
<b>Deskripsi</b> :	
➤ Input	: alpha[i], alpha[j]
➤ Proses:	
	a) Mendapatkan nilai alpha[i] dan alpha[j]
	b) Menginputkan nilai alpha[i] dan alpha[j] ke dalam variabel ai_lama dan aj_lama
	c) Menampilkan hasil ai_lama dan aj_lama

Gambar 4.47 *Pseudocode* Men-*set* Nilai Alpha

Sumber: [Perancangan]

Hasil men-*set* nilai alpha ke dalam variabel alpha\_lama ditunjukkan pada tabel 4.21.

Tabel 4.21 Nilai Alpha data *i* dan *j*

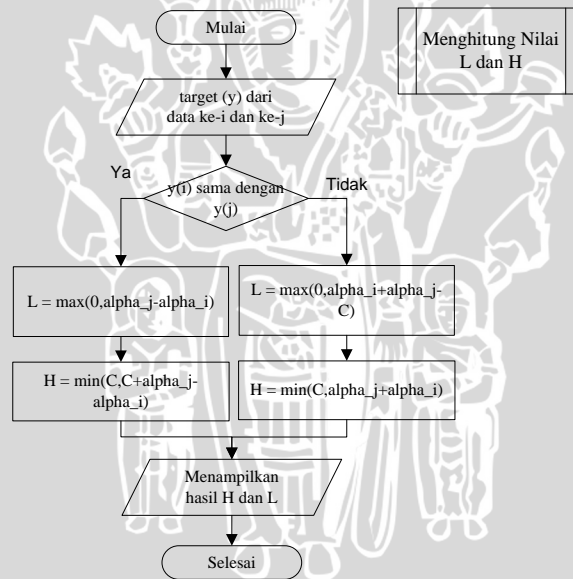
Perulangan ke-	i	j	Alpha_lama-i	Alpha_lama-j
1	1	3	0	0
2	2	6	0	0
3	3	6	0	1

4	4	5	0	0
5	null	null	null	null
6	null	null	null	null
7	1	5	0	1
8	2	4	1	1
9	3	5	0	1
10	null	null	null	null
11	null	null	null	null
12	null	null	null	null

Sumber: [Perancangan]

• Tahap ke – 5

Menghitung segmen garis diagonal dengan mencari nilai L dan H sesuai dengan persamaan (2-32) dan (2-33). Pencarian nilai L dan H ditentukan dari target yang dimiliki dari data *i* dan data *j*. Diagram alir untuk mencari nilai L dan H dapat dilihat pada gambar 4.48.



Gambar 4.48 Diagram Alir Menghitung Segmen Garis Diagonal

Sumber: [Perancangan]

Pseudocode proses menghitung segmen garis diagonal sebagai berikut :

<p><b>Nama Pseudocode</b> : menghitung segmen garis diagonal</p> <p><b>Deklarasi</b> :</p> <ul style="list-style-type: none"> <li>➤ Double : L, H, alpha[i], alpha[j]</li> </ul> <p><b>Deskripsi</b> :</p> <ul style="list-style-type: none"> <li>➤ Input : y[i], y[j], alpha[i], alpha[j]</li> <li>➤ Proses:                     <ol style="list-style-type: none"> <li>a) Mengecek target(y) pada kedua data (i dan j).</li> <li>b) Jika kedua target sama maka akan dilakukan proses</li> </ol> </li> </ul>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



perhitungan nilai L dan H sesuai dengan persamaan yang telah ditentukan begitu juga sebaliknya  
c) Menampilkan hasil L dan H

Gambar 4.49 Pseudocode Menghitung Segmen Garis Diagonal  
Sumber: [Perancangan]

Contoh perhitungan mencari nilai L dan H berdasarkan perhitungan pada tahap 1 dan 3 target yang didapatkan adalah  $y[i] = 1$  dan  $y[j] = 1$  (sama).

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

$$L = \max(0, 0 - 0 - 1) \quad H = \min(1, 0 + 0)$$

$$L = 0 \quad H = 0$$

Berdasarkan perhitungan di atas maka didapatkan nilai L dan H pada tiap data latih untuk masing-masing perulangan seperti yang ditunjukkan pada tabel 4.22 dan 4.23.

Tabel 4.22 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

Sumber: [Perancangan]

Tabel 4.23 Nilai L dan H (target tidak sama)

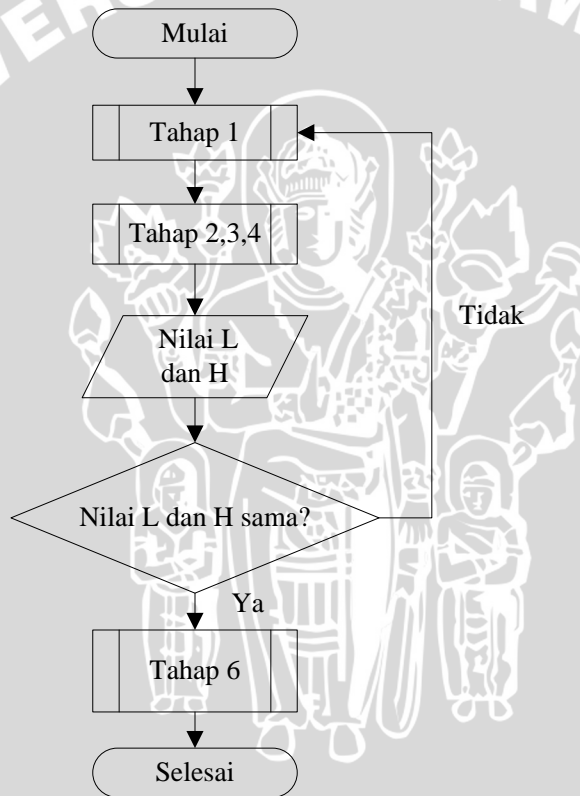
Perulangan ke-	L	H	target i	target j
1	0	1	1	1
2	0	1	1	-1
3	1	1	1	-1
4	0	1	1	-1
5	null	null	null	null
6	null	null	null	null

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

Sumber: [Perancangan]

- Tahap ke – 6

Mengecek hasil dari perhitungan L dan H sama atau tidak, jika sama maka kembali ke tahap 1 dengan i++. Diagram alir untuk mengecek nilai L dan H dapat dilihat pada gambar 4.50.



Gambar 4.50 Diagram Alir Mengecek Nilai L dan H  
 Sumber: [Perancangan]

Pseudocode proses mengecek nilai L dan H sebagai berikut :

<p><b>Nama Pseudocode</b> : mengecek nilai L dan H</p> <p><b>Deklarasi</b> :</p> <ul style="list-style-type: none"> <li>➤ Double : L, H</li> </ul> <p><b>Deskripsi</b> :</p> <ul style="list-style-type: none"> <li>➤ Input : L,H</li> <li>➤ Proses:                     <ol style="list-style-type: none"> <li>Mengecek nilai L dan H apakah sama atau tidak</li> </ol> </li> </ul>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

b) Jika nilai keduanya sama maka proses akan terhenti dan akan kembali ke tahap 1, jika tidak akan lanjut ke tahap selanjutnya

Gambar 4.51 Pseudocode Mengecek Nilai L dan H  
Sumber: [Perancangan]

Hasil pengecekan terhadap nilai L dan H ditunjukkan pada tabel 4.24.

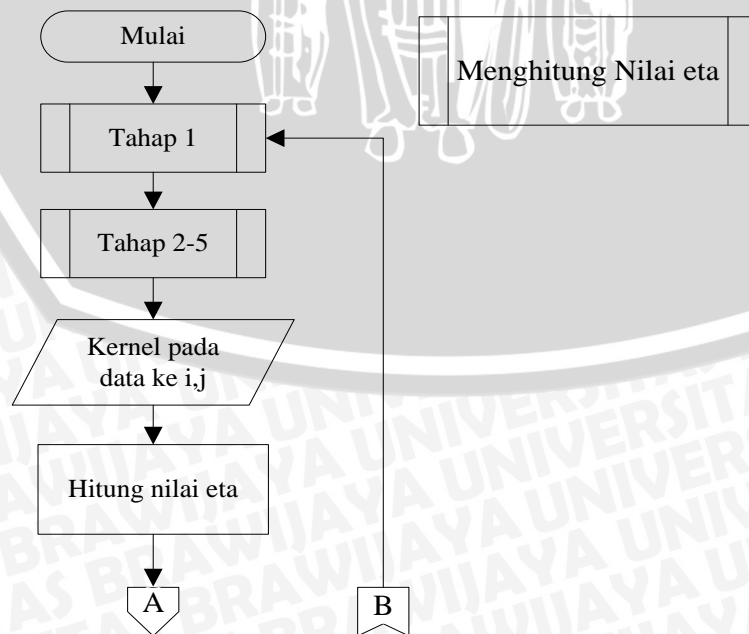
Tabel 4.24 Nilai L dan H (Kondisi)

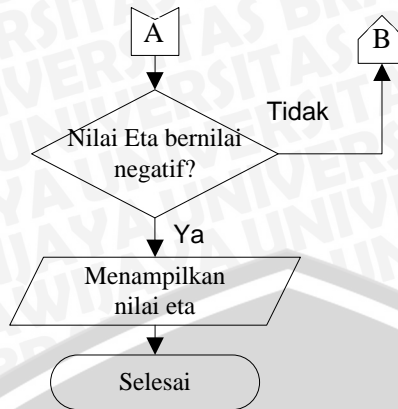
Perulangan ke-	Hasil	L	H	Target(y) i	Target(y) j
1	SAMA	0	0	1	1
2	TIDAK SAMA	0	1	1	-1
3	SAMA	1	1	1	-1
4	TIDAK SAMA	0	1	1	-1
5	null	null	null	null	null
6	null	null	null	null	null
7	SAMA	1	1	1	-1
8	SAMA	1	1	1	1
9	SAMA	1	1	1	1
10	null	null	null	null	null
11	null	null	null	null	null
12	null	null	null	null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

Sumber: [Perancangan]

- Tahap ke – 7

Menghitung nilai eta ( $\eta$ ) dengan persamaan (2-37) dan mengecek lokasi batas maksimal dari fungsi objektif. Diagram alir untuk menghitung nilai eta pada masing-masing proses perulangan dapat dilihat pada gambar 4.52.





Gambar 4.52 Diagram Alir Menghitung *Eta*  
 Sumber: [Perancangan]

*Pseudocode* proses menghitung nilai *eta* sebagai berikut :

**Nama Pseudocode** : menghitung nilai *eta*  
**Deklarasi** :  
 ➤ Double :  $K(i, j), K(j, j), K(i, i), eta$   
**Deskripsi** :  
 ➤ Input :  $K(i, j), K(j, j), K(i, i)$   
 ➤ Proses:  
 a) Mendapatkan nilai dot product dari fungsi kernel  $K(i, j), K(j, j), K(i, i)$ .  
 b) Menghitung nilai *eta* dengan menggunakan persamaan yang telah ditentukan.  
 c) Mengecek nilai *eta* yang didapatkan apakah bernilai positif, jika bernilai positif maka akan kembali ke tahap 1 dengan  $(i++)$ .  
 d) Menampilkan nilai *eta*

Gambar 4.53 *Pseudocode* Menghitung Nilai *Eta*  
 Sumber: [Perancangan]

Contoh perhitungan *eta* ( $\eta$ ) pada perulangan ke-4 untuk  $i = 4$  dan  $j = 5$  dengan menggunakan persamaan (2-37). Nilai  $K(x_i, x_j), K(x_i, x_i), K(x_j, x_j)$  didapatkan pada perhitungan menggunakan kernel RBF pada tabel 4.17.

$$\begin{aligned}
 \eta &= 2K(x_i, x_j) - K(x_i, x_i) - K(x_j, x_j) \\
 &= 2 * 0,841979173 - 1,00 - 1,00 \\
 &= -3,16041653663000E - 01
 \end{aligned}$$

Berdasarkan contoh perhitungan di atas maka hasil perhitungan nilai *eta* pada masing-masing proses ditunjukkan pada tabel 4.25.

Tabel 4.25 Hasil perhitungan *eta*

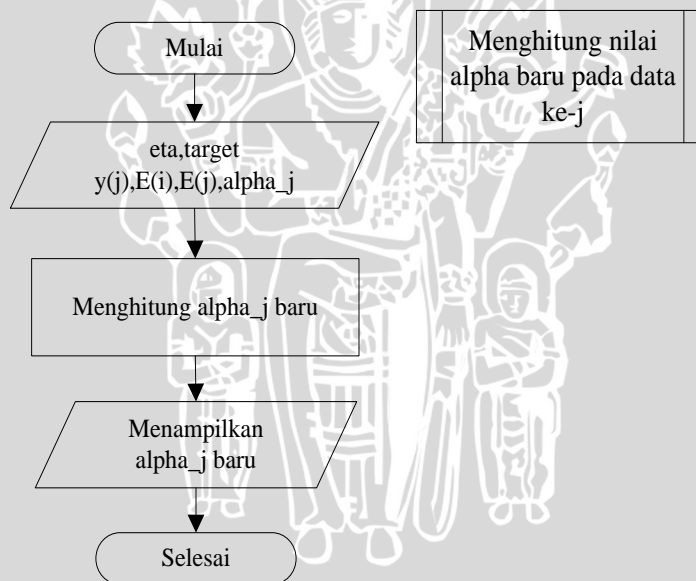
Perulangan ke-	Eta	i	j
1	null	1	3



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

Sumber: [Perancangan]

- Tahap ke – 8  
Menghitung nilai alpha baru pada data ke-j dengan menggunakan persamaan (2-34). Diagram alir untuk menghitung nilai alpha baru pada data ke-j pada masing-masing proses perulangan dapat dilihat pada gambar 4.54.



Gambar 4.54 Diagram Alir Menghitung Nilai Alpha Baru Pada Data ke-j  
Sumber: [Perancangan]

Pseudocode proses menghitung alpha baru pada data ke-j sebagai berikut :

```

Nama Pseudocode : menghitung alpha baru pada data ke-j
Deklarasi :
    ➤ Double : y[j], eta
Deskripsi :
    ➤ Input : y[j], eta, alpha[i], alpha[j], E[i], E[j]
    ➤ Proses:
        a) Mendapatkan nilai eta, target dan nilai alpha.
    
```



- b) Menghitung nilai alpha pada data ke-j dengan menggunakan persamaan yang telah ditentukan.  
 c) Menampilkan nilai alpha baru pada data ke-j

Gambar 4.55 Pseudocode Menghitung Nilai Alpha Baru Pada Data ke-j  
 Sumber: [Perancangan]

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 dari perhitungan tahap 1 dan tahap 3 pada tabel 4.18 dan tabel 4.20.

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

Berdasarkan contoh perhitungan di atas maka hasil perhitungan nilai *alpha\_baru* pada masing-masing proses ditunjukkan pada tabel 4.26.

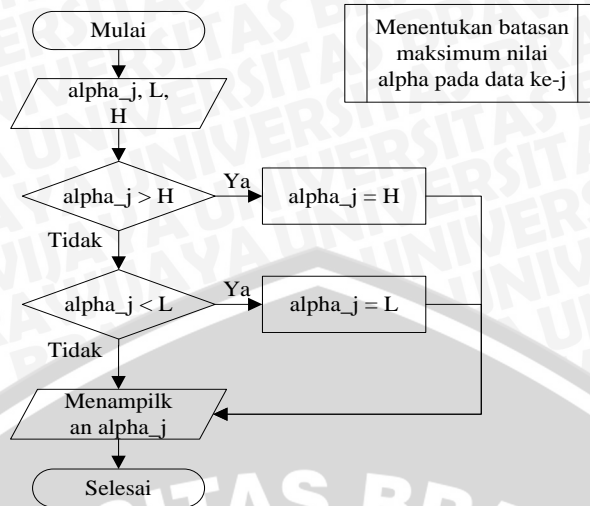
Tabel 4.26 Nilai Alpha Baru data ke-j

Perulangan ke-	alpha_baru ke-j	i	j
1	null	1	3
2	1,00000000000066900E+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

Sumber: [Perancangan]

- Tahap ke – 9

Menentukan batasan maksimum nilai alpha dengan memotong segmen garis selain batas maksimum. Diagram alir untuk menentukan batasan maksimum nilai alpha pada data ke-j pada masing-masing proses perulangan dapat dilihat pada gambar 4.56.



Gambar 4.56 Diagram Alir Menentukan Batasan Maksimum Nilai Alpha  
 Sumber: [Perancangan]

*Pseudocode* proses menentukan batasan maksimum nilai alpha sebagai berikut :

**Nama Pseudocode** : menentukan batasan maksimum nilai alpha  
**Deklarasi** :  
 ➤ Double : L, H  
**Deskripsi** :  
 ➤ Input : L,H, alpha\_baru[j]  
 ➤ Proses:  
 a) Mendapatkan nilai L, H dan alpha baru data ke-j  
 b) Mengecek nilai alpha baru apakah lebih besar dari H, jika lebih besar maka nilai alpha baru = nilai H.  
 c) Mengecek nilai alpha baru lebih kecil dari L jika lebih kecil maka nilai alpha baru = nilai L.  
 d) Jika tidak memenuhi kedua batasan tersebut maka nilai alpha tidak ada perubahan.  
 e) Menampilkan nilai alpha j

Gambar 4.57 *Pseudocode* Menentukan Batasan Maksimum Nilai Alpha  
 Sumber: [Perancangan]

Hasil menentukan batasan maksimum nilai alpha pada masing-masing perulangan ditunjukkan pada tabel 4.27.

Tabel 4.27 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

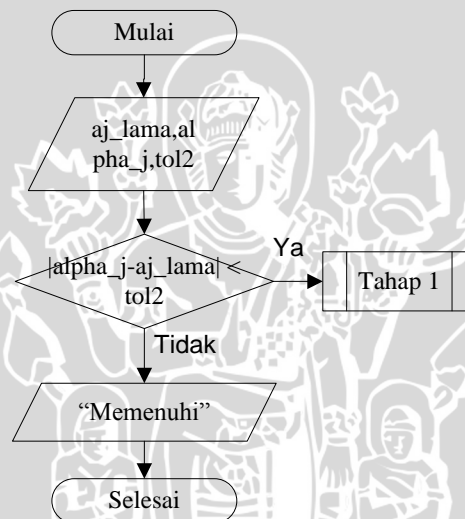


6	null	null
7	null	null
8	null	null
9	null	null
10	null	null
11	null	null
12	null	null

Sumber: [Perancangan]

- Tahap ke – 10

Mengecek nilai absolut dari  $alpha\_baru[j] - alpha\_lama[j]$  lebih kecil dari nilai tol2 terpenuhi atau tidak, jika terpenuhi maka akan kembali ke tahap 1. Diagram alir untuk mengecek kondisi tersebut dapat dilihat pada gambar 4.58.



Gambar 4.58 Diagram Alir Mengecek Nilai Absolut Alpha Baru Data ke-*j*  
Sumber: [Perancangan]

*Pseudocode* proses mengecek nilai absolut alpha baru data ke-*j* sebagai berikut :

<p><b>Nama Pseudocode</b> : mengecek nilai absolut alpha baru data ke-<i>j</i></p> <p><b>Deklarasi</b> :</p> <ul style="list-style-type: none"> <li>➤ Double : tol2, alpha_lama[j]</li> </ul> <p><b>Deskripsi</b> :</p> <ul style="list-style-type: none"> <li>➤ Input : alpha_lama[j], alpha[j], tol2</li> <li>➤ Proses:           <ol style="list-style-type: none"> <li>a) Mendapatkan alpha lama[j], alpha[j] dan tol2</li> <li>b) Mengecek nilai absolut dari <math>alpha[j] - alpha\_lama[j] &lt; tol2</math> ,jika lebih kecil maka akan kembali ke tahap 1.</li> </ol> </li> </ul>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Gambar 4.59 *Pseudocode* Mengecek Nilai Absolut Alpha Baru Data ke-*j*  
Sumber: [Perancangan]



Hasil dari mengecek nilai absolut pada masing-masing proses ditunjukkan pada tabel 4.28.

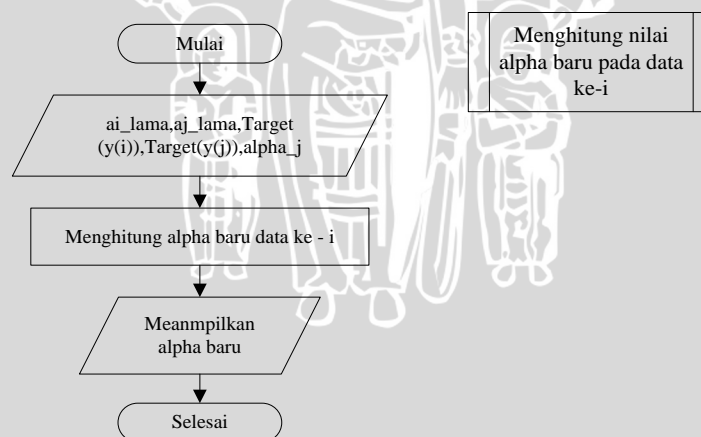
Tabel 4.28 Kondisi nilai absolut

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

Sumber: [Perancangan]

- Tahap ke – 11

Menghitung nilai alpha baru pada data ke- $i$  menggunakan persamaan (2-39). Diagram alir untuk menghitung nilai alpha baru pada data ke- $i$  dapat dilihat pada gambar 4.60.



Gambar 4.60 Diagram Alir Menghitung Nilai Alpha Data ke -  $i$   
Sumber: [Perancangan]

*Pseudocode* proses menghitung alpha baru data ke- $i$  sebagai berikut :

<p><b>Nama Pseudocode</b> : menghitung alpha baru data ke-<math>i</math>  <b>Deklarasi</b> :          ➤ Double : alpha_lama[j], alpha_lama[i]  <b>Deskripsi</b> :          ➤ Input : alpha_lama[j], alpha_lama[i], alpha[j], y[i], y[j]</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

➤ Proses:

- a) Mendapatkan nilai alpha lama[j], alpha lama[i], y[i], y[j] dan alpha[j]
- b) Menghitung alpha baru menggunakan persamaan yang telah ditentukan.
- c) Menampilkan alpha baru pada data ke-i

Gambar 4.61 Pseudocode Menghitung Nilai Alpha Data ke - i

Sumber: [Perancangan]

Contoh perhitungan *alpha baru* data ke-i pada perulangan ke-4 untuk  $i = 4$  dan  $j = 5$ . Nilai  $\alpha_j$  didapatkan dari perhitungan tahap ke-9 pada tabel 4.27.

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

Berdasarkan perhitungan diatas didapatkan hasil  $\alpha_i$  pada masing-masing perulangan ditunjukkan pada tabel 4.29.

Tabel 4.29 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

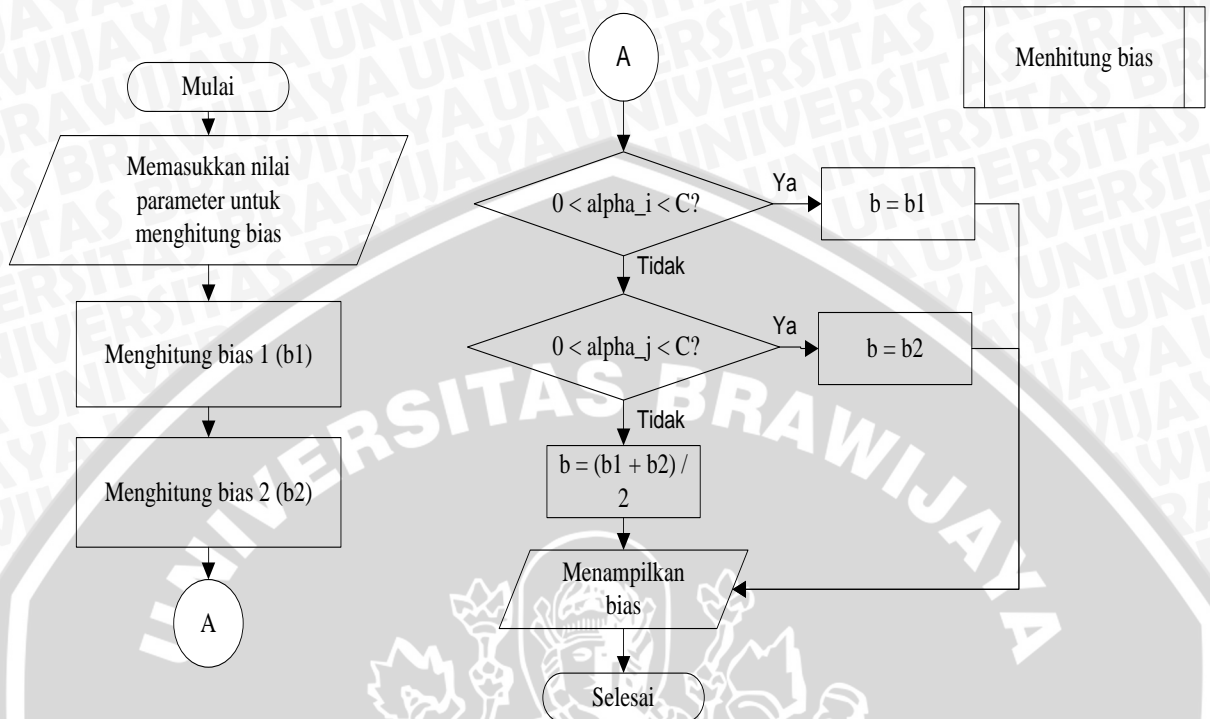
Sumber: [Perancangan]

- Tahap ke – 12

Menghitung nilai bias ( $b_1$  dan  $b_2$ ) menggunakan persamaan (2-40) dan (2-41) dengan memenuhi beberapa kondisi antara lain:

- a. Kondisi 1 : jika memenuhi  $0 < \alpha_i < C$  maka  $b_1$  digunakan.
- b. Kondisi 2 : jika memenuhi kondisi  $0 < \alpha_j < C$  maka  $b_2$  digunakan.

c. Kondisi 3 :jika tidak memenuhi kondisi diatas maka nilai bias setengah dari total  $b_1$  dan  $b_2$ .



Gambar 4.62 Diagram Alir Menghitung Nilai Bias  
Sumber: [Perancangan]

Pseudocode proses menghitung nilai bias sebagai berikut :

**Nama Pseudocode** : menghitung nilai bias

**Deklarasi** :

- Double : bias, C, K(i,i),K(i,j),K(j,j)

**Deskripsi** :

- Input : alpha\_lama[j],alpha\_lama[i],alpha[j],y[i],y[j],K(i,i),K(j,j),K(i,j)
- Proses:
  - a) Memasukan parameter yang dibutuhkan dalam proses perhitungan bias
  - b) Menghitung nilai b1 dan b2
  - c) Dilakukan pengecekan terhadap alpha[i] dan alpha[j] terhadap batasan nilai C.
  - d) Jika  $\alpha[i] > 0$  dan  $\alpha[i] < C$  maka nilai bias = b1
  - e) Jika  $\alpha[j] > 0$  dan  $\alpha[j] < C$  maka nilai bias = b2
  - f) Jika tidak memenuhi kondisi diatas maka nilai bias =  $(b1 + b2) / 2$
- Output : alpha[i]

Gambar 4.63 Pseudocode Menghitung Nilai Alpha Data ke - i  
Sumber: [Perancangan]



Contoh perhitungan *bias* data ke-*i* pada perulangan ke-4 untuk  $i = 4$  dan  $j = 5$ . Nilai  $E_i$  dan  $E_j$  didapatkan dari perhitungan tahap 1 dan tahap 3 pada tabel 4.18 dan tabel 4.20. Nilai  $K(x_i, x_j)$ ,  $K(x_i, x_i)$ ,  $K(x_j, x_j)$  didapatkan pada perhitungan menggunakan kernel RBF pada tabel 4.17.

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

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

Tabel 4.30 Nilai b1 dan b2

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

Sumber: [Perancangan]

Setelah mendapatkan nilai pada bias\_1 dan bias\_2 maka akan dilakukan perhitungan untuk mengitung bias optimal (bias\_baru) sesuai dengan persamaan (2-42) pada tiap perulangan. Nilai  $\alpha_j$  dan  $\alpha_i$  masing-masing dapat dilihat pada tabel 4.27 dan 4.29. Berikut contoh perhitngan 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.

$$b = \frac{b1 + b2}{2}$$

$$= \frac{0,984823389 + (-0,731838861)}{2}$$

$$= 0,126492264$$

Tabel 4.31 Nilai bias akhir

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

Sumber: [Perancangan]

Setelah semua data latih tidak memenuhi kondisi KKT (tahap ke 1) maka didapatkan nilai optimal dari *bias* dan *alpha* pada masing-masing data latih. Tabel 4.32 menunjukkan nilai *bias* dan *alpha* optimal dari proses pelatihan pertama (Baik vs Buruk). Untuk mendapatkan nilai alpha pada tiap data latih dapat dilihat pada tabel 4.27 dan 4.29.

Tabel 4.32 Nilai Alpha dan Bias Pada Pelatihan Pertama

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

Sumber: [Perancangan]

Berdasarkan nilai  $\alpha$  dan  $b$  dari pelatihan pertama (Baik vs Buruk) maka didapatkan fungsi keputusan yang sesuai dengan persamaan (2-15) untuk pelatihan diatas adalah sebagai berikut:

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

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

### Proses Pelatihan Data Latih Kedua (Baik vs Sedang)

Tahap selanjutnya adalah melakukan pelatihan kedua pada data training. Data yang digunakan hanya kelas baik dan sedang dari keseluruhan data training. Proses pelatihan ini tahapan dan parameter yang digunakan sama seperti proses pelatihan pertama (baik vs sedang).

Tabel 4.33 Data Pelatihan Kedua

Data Gambar ke-	Id Gambar	Fitur			Kelas	Keterangan Kelas	Target (y)
		R	G	B			
X1	SB-2	70	67	55	1	Baik	1
X2	SB-5	57	61	68	1	Baik	1
X3	SB-17	133	88	50	2	Sedang	-1
X4	SB-20	164	124	96	2	Sedang	-1

Sumber: [Perancangan]

Tabel 4.33 terdapat 4 data latih untuk tiap kelas Baik dan Sedang. Pada pelatihan kedua data gambar kelas Baik memiliki target (y) +1 sedangkan data gambar kelas sedang memiliki target (y) -1. Berikut adalah proses pelatihan kedua untuk kelas Baik dan Sedang.

- Tahap ke – 1

Mencari nilai  $error$  pada setiap data training (data ke- $i$ ) dengan menggunakan persamaan (2-36). Tabel 4.34 menunjukkan hasil  $error$  pada data ke-  $i$  dan setiap perulangan. Nilai target ( $y_i$ ) didapatkan pada tabel 4.33.

Contoh perhitungan nilai *error* ( $E_i$ ) pada data latih adalah misal mencari  $error(i)$  pada perulangan ke-1 dengan data ke-1,  $\alpha = 0$ , bias = 0, target( $y$ ) = 1

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

$$E_i = f(x) - y_i$$

$$f(x) = ((0 * 1 * 1) + (0 * 1 * 0,687976912) + (0 * 1 * 0,011855066) + (0 * 1 * 1,05076E - 06) + 0 = 0$$

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

Tabel 4.34 Hasil *Error* pada data ke-i

Perulangan ke-	data ke-i	f(x)	yi	Ei
1	1	0	1	-1
2	2	0,68797682	1	-0,31202318
3	3	-1	-1	0
4	4	-1,351886834	-1	-0,351886834
5	1	1	1	0
6	2	0,970366719	1	-0,029633281
7	3	-0,859318027	-1	0,140681973
8	4	-1,083584248	-1	-0,083584248
9	1	1	1	0
10	2	0,97039785	1	-0,02960215
11	3	-0,943886143	-1	0,056113857
12	4	-1	-1	0
13	1	0,964294568	1	-0,035705432
14	2	0,994782704	1	-0,005217296
15	3	-1	-1	0
16	4	-0,982375149	-1	0,017624851
17	1	1,001653646	1	0,001653646
18	2	1	1	0
19	3	-1,006335478	-1	-0,006335478
20	4	-1	-1	0

Sumber: [Perancangan]

- Tahap ke – 2

Melakukan pengecekan terhadap kondisi *Karush-Kuhn-Tucker* (KKT) pada data training dan nilai  $E_i$ , jika tidak memenuhi kondisi tersebut maka perulangan kembali pada tahap ke – 1 dengan melakukan perhitungan error

pada data ke-i selanjutnya. Tabel 4.35 menunjukkan hasil kondisi terhadap KKT pada tiap perulangan.

Tabel 4.35 Kondisi KKT

Perulangan ke-	Hasil	Perulangan ke-	Hasil
1	True	11	True
2	True	12	False
3	False	13	True
4	True	14	False
5	False	15	False
6	True	16	True
7	True	17	False
8	True	18	False
9	False	19	False
10	True	20	False

Sumber:[Perancangan]

- Tahap ke – 3

Tahap berikutnya adalah mencari data pembanding dari data latih secara *random* (acak) dengan ketentuan  $j \neq i$  dan mencari nilai *error* ( $E_j$ ) pada data tersebut. Tabel 4.36 menunjukkan data pembanding yang didapatkan secara acak beserta nilai error ( $E_j$ ).

Proses perhitungan nilai *error* pada data ke-j sama seperti perhitungan nilai *error* pada data ke-i menggunakan persamaan (2-36), tetapi untuk menentukan data pembanding dilakukan secara acak. Berikut contoh perhitungan nilai *error*(j) pada perulangan ke-1, misalkan nilai  $j = 4$ .

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

$$E_j = f(x) - y_j$$

$$f(x) = ((0 * 1 * 0,011855066) + (0 * 1 * 0,00108194) + (0 * 1 * 1) + (0 * 1 * 0,012613344) + 0 = 0$$

$$E_4 = 0 - 1 = -1$$

Tabel 4.36 *Error* data pembanding ( $E_j$ )

Perulangan ke-	data ke-j	f(x)	yj	Ej
1	4	0	-1	1
2	3	-0,00076	-1	0,999242
3	null	null	null	null
4	1	1,100157	1	0,100157



5	null	null	null	null
6	4	-1	-1	0
7	1	1,02501	1	0,02501
8	1	1	1	0
9	null	null	null	null
10	1	1	1	0
11	4	-0,9852	-1	0,014801
12	null	null	null	null
13	3	-1	-1	0
14	null	null	null	null
15	null	null	null	null
16	2	0,994783	1	-0,00522
17	null	null	null	null
18	null	null	null	null
19	null	null	null	null
20	null	null	null	null

Sumber:[Perancangan]

- Tahap ke – 4

Men-*set* nilai alpha pada data *i* dan data *j* ke dalam variabel *alpha\_lama*.

Tabel 4.37 menunjukkan nilai alpha pada variabel *alpha\_lama*. Nilai alpha lama tiap perulangan dapat dilihat pada tabel 4.43 dan 4.45.

Tabel 4.37 Nilai Alpha data *i* dan *j*

Perulangan ke-	<i>i</i>	<i>j</i>	<i>alpha_lama-i</i>	<i>alpha_lama-j</i>
1	1	4	0	0
2	2	3	0	0
3	null	null	null	null
4	4	1	1	1
5	null	null	null	null
6	2	4	0,656342574	0,77397773
7	3	1	0,656342574	0,77397773
8	4	1	0,788794372	0,832507525
9	null	null	null	null
10	2	1	0,671159216	0,790715357
11	3	4	0,714872369	0,747002204
12	null	null	null	null
13	1	3	0,743279525	0,735792657
14	null	null	null	null
15	null	null	null	null
16	4	2	0,726081915	0,718595048
17	null	null	null	null
18	null	null	null	null

19	null	null	null	null
20	null	null	null	null

Sumber: [Perancangan]

- Tahap ke – 5

Menghitung segmen garis diagonal dengan mencari nilai L dan H sesuai dengan persamaan (2-32) dan (2-33). Pencarian nilai L dan H ditentukan dari target yang dimiliki dari data  $i$  dan data  $j$ . Tabel 4.38 menunjukkan nilai L dan H jika target pada kedua data SAMA. Tabel 4.39 menunjukkan nilai L dan H jika target pada kedua data TIDAK SAMA

Contoh perhitungan mencari nilai L dan H. Berdasarkan perhitungan pada tahap 1 dan 3 target yang didapatkan adalah  $y[i] = 1$  dan  $y[j] = -1$  (tidak sama).

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

$$L = \max(0, 0 - 0 - 1) \quad H = \min(1, 1 + 0 + 0)$$

$$L = 0 \quad H = 1$$

Tabel 4.38 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	null	null	null	null
4	1	1	-1	1
5	null	null	null	null
6	0,430320304	1	1	-1
7	0,430320304	1	-1	1
8	0,621301896	1	-1	1
9	null	null	null	null
10	0,461874573	1	1	1
11	0,461874573	1	-1	-1
12	null	null	null	null
13	0,479072182	1	1	-1
14	null	null	null	null
15	null	null	null	null
16	0,444676964	1	-1	1
17	null	null	null	null
18	null	null	null	null

19	null	null	null	null
20	null	null	null	null

Sumber: [Perancangan]

Tabel 4.39 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	null	null	null	null
4	0	1	-1	1
5	null	null	null	null
6	0,117635156	1	1	-1
7	0,117635156	1	-1	1
8	0,043713153	1	-1	1
9	null	null	null	null
10	0,119556141	1	1	1
11	0,032129835	1	-1	-1
12	null	null	null	null
13	0	0,992513133	1	-1
14	null	null	null	null
15	null	null	null	null
16	0	0,992513133	-1	1
17	null	null	null	null
18	null	null	null	null
19	null	null	null	null
20	null	null	null	null

Sumber: [Perancangan]

- Tahap ke – 6

Mengecek hasil dari perhitungan L dan H sama atau tidak, jika sama maka kembali ke tahap 1 dengan  $i++$ . Tabel 4.40 menunjukkan perulangan yang memiliki nilai L dan H yang sama atau tidak.

Tabel 4.40 Nilai L dan H (Kondisi)

Perulangan ke-	Hasil	L	H	Target(y) i	Target(y) j
1	TIDAK SAMA	0	1	1	-1
2	TIDAK SAMA	0	1	1	-1
3	null	null	null	null	null
4	TIDAK SAMA	0	1	-1	1
5	null	null	null	null	null
6	TIDAK SAMA	0,117635156	1	-1	1
7	TIDAK SAMA	0,117635156	1	1	-1

8	TIDAK SAMA	0,043713153	1	-1	1
9	null	null	null	null	null
10	TIDAK SAMA	0,461874573	1	1	1
11	TIDAK SAMA	0,461874573	1	1	1
12	null	null	null	null	null
13	TIDAK SAMA	0	0,99251313	1	-1
14	null	null	null	null	null
15	null	null	null	null	null
16	TIDAK SAMA	0	0,99251313	-1	1
17	null	null	null	null	null
18	null	null	null	null	null
19	null	null	null	null	null
20	null	null	null	null	null

Sumber: [Perancangan]

- Tahap ke – 7

Menghitung nilai eta ( $\eta$ ) dengan persamaan (2-37) dan mengecek lokasi batas maksimal dari fungsi objektif. Tabel 4.41 menunjukkan hasil dari perhitungan nilai *eta* pada masing-masing perulangan.

Contoh perhitungan *eta* ( $\eta$ ) pada perulangan ke-1 untuk  $i = 1$  dan  $j = 4$  dengan menggunakan persamaan (2-37). Nilai  $K(x_i, x_j)$ ,  $K(x_i, x_i)$ ,  $K(x_j, x_j)$  didapatkan pada perhitungan menggunakan kernel RBF pada tabel 4.17.

$$\begin{aligned}\eta &= 2K(x_i, x_j) - K(x_i, x_i) - K(x_j, x_j) \\ &= (2 * 1,05076E - 06) - 1,00 - 1,00 \\ &= -1,999997898\end{aligned}$$

Tabel 4.41 Hasil perhitungan *eta*

Perulangan ke-	eta	i	j
1	-1,999997898	1	4
2	-1,997836121	2	3
3	null	null	null
4	-1,999997898	4	1
5	null	null	null
6	-1,999999816	2	4
7	-1,976289868	3	1
8	-1,999997898	4	1
9	null	null	null
10	-0,624046176	2	1
11	-1,974773313	3	4
12	null	null	null

13	-1,976289868	1	3
14	null	null	null
15	null	null	null
16	-1,999999816	4	2
17	null	null	null
18	null	null	null
19	null	null	null
20	null	null	null

Sumber: [Perancangan]

- Tahap ke – 8

Menghitung nilai alpha baru pada data ke-j dengan menggunakan persamaan (2-25). Tabel 4.42 menunjukkan hasil alpha\_baru pada data  $j$ . Contoh perhitungan *alpha baru* pada perulangan ke-1 untuk  $i = 1$  dan  $j = 4$ . Nilai  $E_i$  dan  $E_j$  didapatkan dari perhitungan tahap 1 dan tahap 3 pada tabel 4.34 dan tabel 4.36.

$$\begin{aligned}\alpha_j^{baru} &= \alpha_j - \frac{y_j(E_i - E_j)}{\eta} \\ &= 0 - \frac{-1(-1 + 1)}{-1,999997898} \\ &= \mathbf{1,000001051}\end{aligned}$$

Tabel 4.42 Nilai Alpha Baru data  $j$

Perulangan ke-	alpha_baru ke-j	i	j
1	1,000001051	1	4
2	0,656342574	2	3
3	null	3	null
4	0,77397773	4	1
5	null	1	null
6	0,788794372	2	4
7	0,832507525	3	1
8	0,790715357	4	1
9	null	1	null
10	0,743279525	2	1
11	0,726081915	3	4
12	null	4	null
13	0,753859558	1	3
14	null	2	null
15	null	3	null
16	0,730016123	4	2

17	null	1	null
18	null	2	null
19	null	3	null
20	null	4	null

Sumber: [Perancangan]

- Tahap ke – 9

Menentukan batasan maksimum nilai alpha dengan memotong segmen garis selain batas maksimum. Contoh perhitungan alpha baru data ke-j pada perulangan ke 1 dengan  $j = 4$ . Diketahui nilai alpha baru data ke-j pada perulangan ke-1 = 1,000001051,  $L = 0$ , dan  $H = 1$ . Nilai alpha baru akan dilakukan perhitungan sesuai dengan persamaan (2-38) sehingga nilai alpha tersebut memenuhi kondisi  $\alpha > H$  maka nilai alpha pada data ke-j = 1.

Tabel 4.43 Nilai Alpha data  $j$  pada segmen garis

Perulangan ke-	alpha ke-j	j
1	1	4
2	0,65634257	3
3	null	null
4	0,77397773	1
5	null	null
6	0,78879437	4
7	0,83250752	1
8	0,79071536	1
9	null	null
10	0,74327952	1
11	0,72608192	4
12	null	null
13	0,75385956	3
14	null	null
15	null	null
16	0,73001612	2
17	null	null
18	null	null
19	null	null
20	null	null

Sumber: [Perancangan]

- Tahap ke – 10

Mengecek nilai absolut dari  $\alpha_{baru}[j] - \alpha_{lama}[j]$  lebih kecil dari nilai  $tol_2$  terpenuhi atau tidak, jika terpenuhi maka akan kembali ke tahap

1. Nilai alpha baru dapat dilihat pada tabel 4.43 dan alpha lama dapat dilihat pada tabel 4.37

Tabel 4.44 Kondisi nilai absolut

Perulangan ke-	hasil	kondisi
1	1	Tidak Memenuhi
2	0,656342574	Tidak Memenuhi
3	null	null
4	0,22602227	Tidak Memenuhi
5	null	null
6	0,014816642	Tidak Memenuhi
7	0,058529795	Tidak Memenuhi
8	0,041792168	Tidak Memenuhi
9	null	null
10	0,047435832	Tidak Memenuhi
11	0,020920288	Tidak Memenuhi
12	null	null
13	0,0180669	Tidak Memenuhi
14	null	null
15	null	null
16	0,011421075	Tidak Memenuhi
17	null	null
18	null	null
19	null	null
20	null	null

Sumber: [Perancangan]

- Tahap ke – 11

Menghitung nilai alpha baru pada data ke- $i$  menggunakan persamaan (2-39). Tabel 4.45 menunjukkan alpha baru pada data ke- $i$ . Contoh perhitungan *alpha baru* data ke- $i$  pada perulangan ke-1 untuk  $i = 1$  dan  $j = 4$ . Nilai  $\alpha_j$  didapatkan dari perhitungan tahap ke-9 pada tabel 4.43.

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

Tabel 4.45 Nilai Alpha Baru Data  $i$

Perulangan ke-	alpha_baru ke-i	i
1	1	1
2	0,656342574	2
3	null	3

4	0,77397773	4
5	null	1
6	0,671159216	2
7	0,714872369	3
8	0,747002204	4
9	null	1
10	0,718595048	2
11	0,735792657	3
12	null	4
13	0,761346425	1
14	null	2
15	null	3
16	0,73750299	4
17	null	1
18	null	2
19	null	3
20	null	4

Sumber: [Perancangan]

- Tahap ke – 12

Menghitung nilai bias ( $b_1$  dan  $b_2$ ) menggunakan persamaan (2-40) dan (2-41) dengan memenuhi beberapa kondisi antara lain:

- Kondisi 1 : jika memenuhi  $0 < \alpha_i < C$  maka  $b_1$  digunakan.
- Kondisi 2 : jika memenuhi kondisi  $0 < \alpha_j < C$  maka  $b_2$  digunakan.
- Kondisi 3 : jika tidak memenuhi kondisi diatas maka nilai bias setengah dari total  $b_1$  dan  $b_2$ .

Contoh perhitungan *bias* data ke-i pada perulangan ke-1 untuk  $i = 1$  dan  $j = 4$ . Nilai  $E_i$  dan  $E_j$  didapatkan dari perhitungan tahap 1 dan tahap 3 pada tabel 4.34 dan tabel 4.36. Nilai  $K(x_i, x_j)$ ,  $K(x_i, x_i)$ ,  $K(x_j, x_j)$  didapatkan pada perhitungan menggunakan kernel RBF pada tabel 4.17.

$$\begin{aligned}
 b_1 &= b - E_i - y_i(\alpha_i - \alpha_i^{lama})K(x_i, x_i) - y_j(\alpha_j - \alpha_j^{lama})K(x_i, x_j) \\
 &= 0 - (-1) - (1 * (1 - 0) * 1) - (-1) * (1 - 0) * 1,05076E - 06 \\
 &= \mathbf{1,05076E - 06}
 \end{aligned}$$

$$b_2 = b - E_i - y_i(\alpha_i - \alpha_i^{lama})K(x_i, x_j) - y_j(\alpha_j - \alpha_j^{lama})K(x_j, x_j)$$



$$= 0 - (-1) - (1 * (1 - 0) * 1,05076E - 06) - (-1) * (1 - 0) * (1)$$

$$= -1,05076E - 06$$

Tabel 4.46 Nilai b1 dan b2

Perulangan ke-	bias_1	bias_2
1	1,05076E-06	-1,05076E-06
2	-0,343609271	-0,343609271
3	null	null
4	-0,217744469	-0,217744469
5	null	null
6	-0,202927829	-0,202927829
7	-0,285773882	-0,285773882
8	-0,243981758	-0,243981758
9	null	null
10	-0,229180683	-0,229180683
11	-0,264638126	-0,264638126
12	null	null
13	-0,24678541	-0,24678541
14	null	null
15	null	null
16	-0,252989187	-0,252989187
17	null	null
18	null	null
19	null	null
20	null	null

Sumber: [Perancangan]

Setelah mendapatkan nilai pada bias\_1 dan bias\_2 maka akan dilakukan perhitungan untuk mengitung bias optimal (bias\_baru) sesuai dengan persamaan (2-42) pada tiap perulangan.

$$b = \frac{b1 + b2}{2}$$

$$= \frac{1,05076E - 06 + (-1,05076E - 06)}{2}$$

$$= -4,86456E - 17$$

Tabel 4.47 Nilai bias akhir

Perulangan ke-	bias_baru
1	-4,86456E-17
2	-0,343609271
3	null
4	-0,217744469
5	null
6	-0,202927829
7	-0,285773882
8	-0,243981758
9	null
10	-0,229180683
11	-0,264638126
12	null
13	-0,24678541
14	null
15	null
16	-0,252989187
17	null
18	null
19	null
20	null

Sumber: [Perancangan]

Setelah semua data latih tidak memenuhi kondisi KKT (tahap ke 1) maka didapatkan nilai optimal dari *bias* dan *alpha* pada masing-masing data latih. Tabel 4.48 menunjukkan nilai *bias* dan *alpha* optimal dari proses pelatihan kedua (Baik vs Sedang). Untuk mendapatkan nilai *alpha* pada tiap data latih dapat dilihat pada tabel 4.43 dan 4.45.

Tabel 4.48 Nilai Alpha dan Bias Pada Pelatihan Kedua

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

Sumber: [Perancangan]

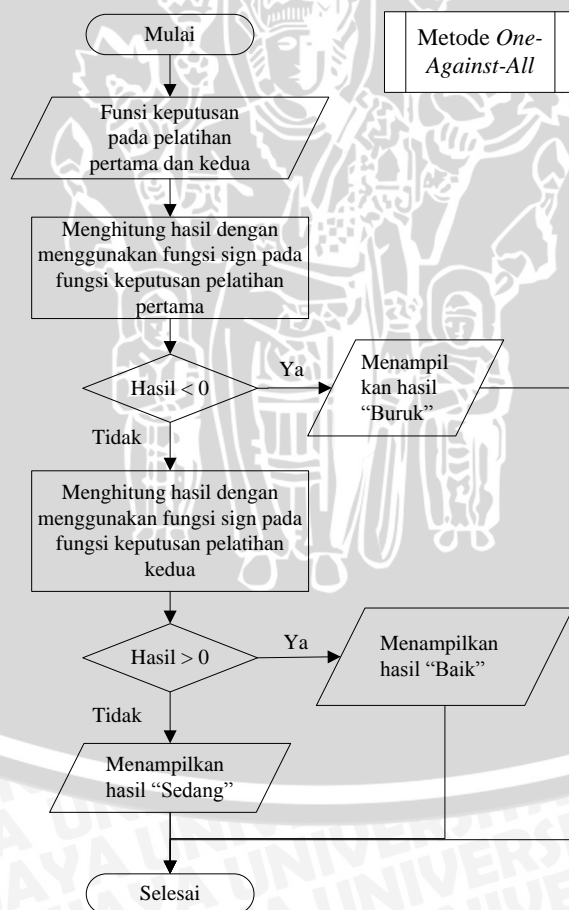
Berdasarkan nilai *alpha* dan *bias* dari pelatihan kedua (Baik vs Sedang) maka didapatkan fungsi keputusan yang sesuai dengan persamaan (2-15) untuk pelatihan diatas adalah sebagai berikut:

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

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

#### 4.2.3.4 Pemodelan Metode *One-Against-All*

Setelah proses pelatihan pada data latih yang dilakukan pada pemodelan algoritma *Simplified SMO*. Proses pelatihan tersebut menghasilkan dua buah fungsi keputusan untuk pelatihan pertama dan pelatihan kedua. Fungsi keputusan ini akan digunakan pada metode *one-against-all* untuk mengatasi penyelesaian klasifikasi *multi-class* (Baik, Sedang dan Buruk) pada penelitian ini. Diagram alir penggunaan algoritma *one-against-all* ditunjukkan pada gambar 4.64.



Gambar 4.64 Diagram Alir Metode *One-Against-All*  
Sumber: [Perancangan]

*Pseudocode* proses klasifikasi menggunakan metode *one-against-all*:

<b>Nama Pseudocode</b>	: klasifikasi <i>one-against-all</i>
<b>Deklarasi</b>	:
➤ double	: hasil
<b>Deskripsi</b>	:
➤ Input	: fitur rata-rata <i>red</i> , <i>green</i> dan <i>blue</i> (RGB)
➤ Proses	:
a)	Mendapatkan fungsi keputusan dari proses pelatihan menggunakan algoritma <i>Simplified SMO</i>
b)	Memasukkan fitur rata-rata RGB dari data uji ke dalam fungsi keputusan.
c)	Menghitung hasil dengan menggunakan fungsi keputusan pada pelatihan pertama.
d)	Jika hasil bernilai $< 0$ maka data uji tersebut diklasifikasikan ke dalam kelas "BURUK".
e)	Jika bernilai $> 0$ akan dilakukan perhitungan hasil dengan menggunakan fungsi keputusan pada pelatihan kedua.
f)	Jika hasil bernilai $> 0$ maka data uji tersebut diklasifikasikan ke dalam kelas "BAIK".
g)	Jika bernilai $< 0$ akan diklasifikasikan ke dalam kelas "SEDANG".

Gambar 4.65 *Pseudocode* Metode *One-Against-All*

Sumber: [Perancangan]

Berikut contoh klasifikasi yang akan dilakukan pada sebuah data uji yang belum diketahui kelasnya menggunakan metode *one-against-all*. Tabel 4.49 merupakan data dari data uji yang akan digunakan dalam proses klasifikasi.

Tabel 4.49 Data Uji

Data Citra ke-	Fitur			Pelatihan Pertama			Pelatihan Kedua		
	R	G	B	Kelas	Ket.Kelas	Target (y)	Kelas	Ket.Kelas	Target (y)
<b>X7</b>	132	123	121	?	?	?	?	?	?

Sumber: [Perancangan]

Tahap awal penggunaan metode *one-against-all* adalah mencari nilai *dot product* dari data uji dengan data latih yang digunakan selama proses pelatihan menggunakan kernel RBF. Tabel 4.50 merupakan nilai *dot product* dari data uji dan data latih menggunakan kernel RBF pada persamaan (2-17).

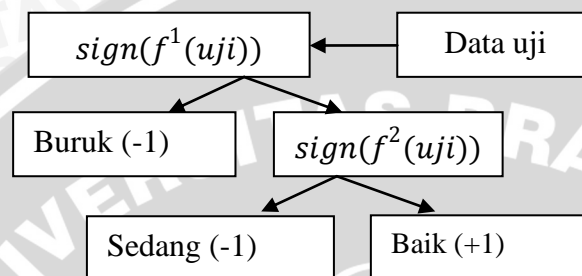
Tabel 4.50 Hasil Perhitungan Kernel RBF Data Uji dan Data Latih

<b>K(xi,xj)</b>	<b>X1</b>	<b>X2</b>	<b>X3</b>	<b>X4</b>	<b>X5</b>	<b>X6</b>
<b>X1</b>	1	0,687976912	0,011855066	1,05076E-06	3,28083E-07	6,67545E-11
<b>X2</b>	0,687976912	1	0,00108194	9,19519E-08	1,76241E-08	6,69273E-12
<b>X3</b>	0,011855066	0,00108194	1	0,012613344	0,01192641	7,1457E-06
<b>X4</b>	1,05076E-06	9,19519E-08	0,012613344	1	0,841979173	0,142844308
<b>X5</b>	3,28083E-07	1,76241E-08	0,01192641	0,841979173	1	0,11014033

<b>X6</b>	6,67545E-11	6,69273E-12	7,1457E-06	0,142844308	0,11014033	1
<b>X7</b>	0,003848776	0,00083007	0,228778727	0,155984287	0,082578989	0,001606014

Sumber: [Perancangan]

Proses klasifikasi *multi-class* terhadap data uji menggunakan metode *One-against-all*. Gambar 4.66 merupakan konsep dasar metode *One-against-all* pada sistem pendukung keputusan pemilihan bibit unggul Sapi Bali.



Gambar 4.66 Diagram Alir Konsep Dasar Metode *One-Against-All*

Sumber: [Perancangan]

Pada gambar 4.66 dijelaskan bahwa jika hasil fungsi keputusan pertama ( $f^1$ ) dari data uji bernilai -1 maka data uji tersebut dapat diklasifikasikan ke dalam kelas Buruk, jika hasilnya (+1) maka akan dilakukan perhitungan pada fungsi keputusan kedua ( $f^2$ ). Apabila hasil dari fungsi keputusan tersebut -1 maka data uji tersebut dapat diklasifikasikan ke dalam kelas Sedang, sedangkan +1 diklasifikasikan ke dalam kelas Baik.

Tahap selanjutnya memasukkan nilai *dot product* untuk data uji ke dalam fungsi keputusan yang telah didapatkan pada proses pelatihan pertama dan pelatihan kedua.

Fungsi keputusan pelatihan pertama:

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

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

Hasil pada fungsi keputusan pertama didapatkan nilai +1 maka dilakukan lagi perhitungan terhadap fungsi keputusan pada pelatihan yang kedua. Berikut hasil fungsi keputusan pada pelatihan yang kedua

Fungsi keputusan pelatihan Kedua:

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

$$f^2(x_{uji}) = \text{sign}(-0,536958878) = -1 \rightarrow \text{Pelatihan Kedua}$$

Berdasarkan hasil dari fungsi keputusan diatas bernilai **-1** maka data uji tersebut diklasifikasikan ke dalam kelas **Sedang**.

#### 4.2.4 Subsistem Antarmuka

Subsistem antarmuka akan membahas rancangan antarmuka dari sistem pendukung keputusan pemilihan bibit unggul Sapi Bali antara lain halaman login, petugas BPTU, olah data pengguna, peternak dan pengujian.

##### 4.2.4.1 Antarmuka Login

Pada antarmuka login berfungsi untuk memberikan hak akses pada setiap pengguna yang ingin masuk ke dalam sistem pendukung keputusan pemilihan bibit unggul Sapi Bali. Rancangan antarmuka login ditunjukkan pada gambar 4.67.

The image shows a login form with the following elements:

- 1: A text input field labeled "Judul Program".
- 2: A square area labeled "Gambar" (Image).
- 3: A text input field labeled "Username".
- 4: A text input field labeled "Password".
- 5: A button (likely "Login").
- 6: A button labeled "Batal" (Cancel).

Gambar 4.67 Antarmuka Login  
Sumber: [Perancangan]

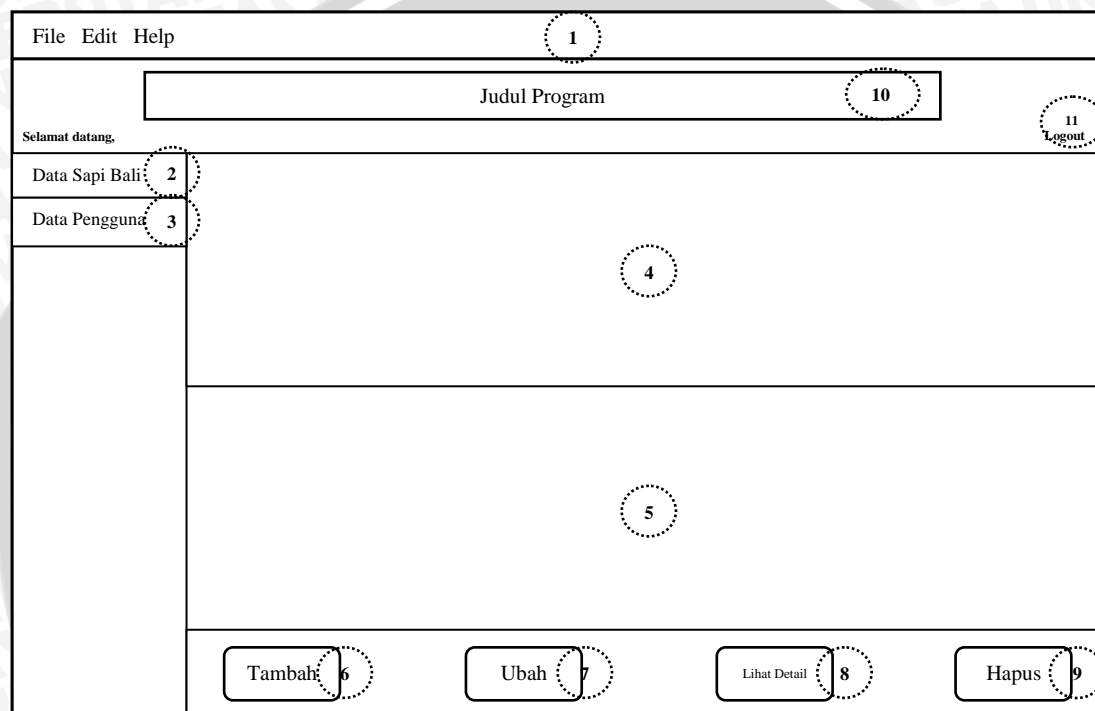
Keterangan gambar 4.67:

1. Judul program
2. Gambar login
3. Input text untuk memasukkakan *username* pengguna

4. Input text untuk memasukkan *password* pengguna
5. Tombol login untuk masuk ke halaman utama sesuai dengan hak akses
6. Tombol batal untuk membatalkan masuk ke dalam sistem

#### 4.2.4.2 Antarmuka Petugas BPTU

Pada antarmuka petugas BPTU terdapat tabpane data Sapi Bali dan data engguna. Rancangan antarmuka petugas BPTU ditunjukkan pada gambar 4.68.



Gambar 4.68 Antarmuka Petugas BPTU  
Sumber: [Perancangan]

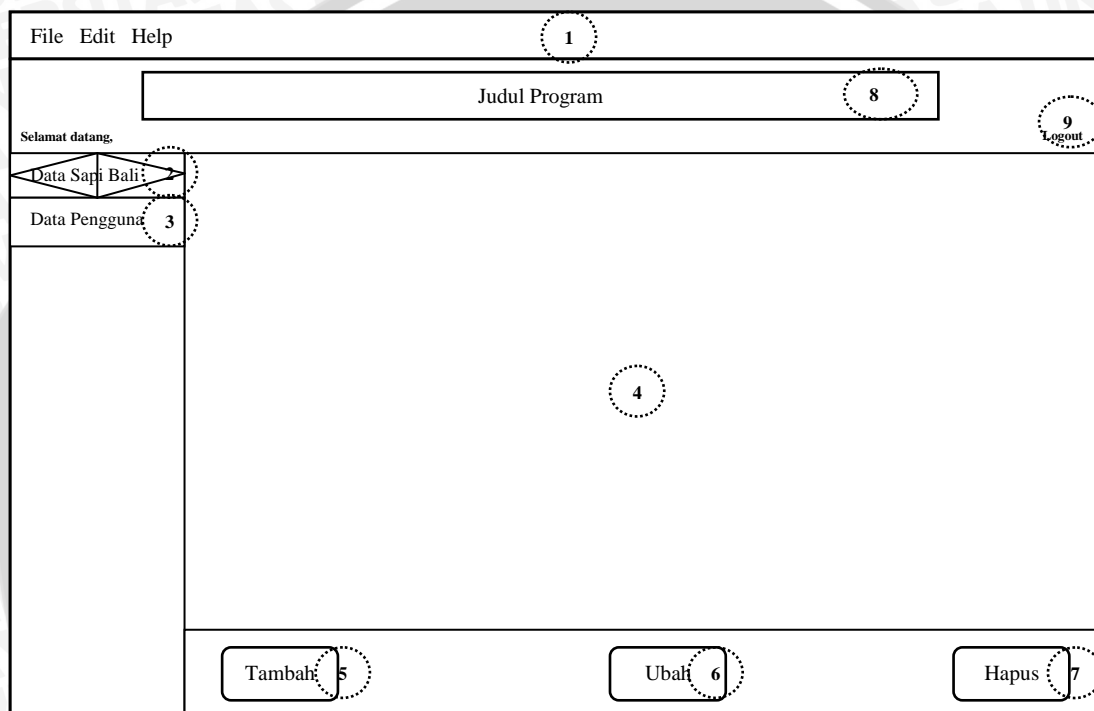
Keterangan gambar 4.68:

1. Menu bar
2. Tabpane untuk data Sapi Bali
3. Tabpane untuk olah data pengguna
4. Tabel untuk data Sapi Bali Betina
5. Tabel untuk data Sapi Bali jantan
6. Tombol tambah untuk menambah data Sapi Bali
7. Tombol ubah untuk mengubah data Sapi Bali
8. Tombol lihat detail untuk mengubah data Sapi Bali
9. Tombol hapus untuk menghapus data Sapi Bali

10. Judul program
11. Tombol logout untuk kembali ke halaman login

#### 4.2.4.3 Antarmuka Olah Data Pengguna

Pada antarmuka olah data pengguna hanya dapat diakses oleh petugas BPTU ketika *username* dan *password* dimasukkan pada halaman login. Rancangan antarmuka olah data pengguna ditunjukkan pada gambar 4.69.



Gambar 4.69 Antarmuka Olah Data Pengguna  
Sumber: [Perancangan]

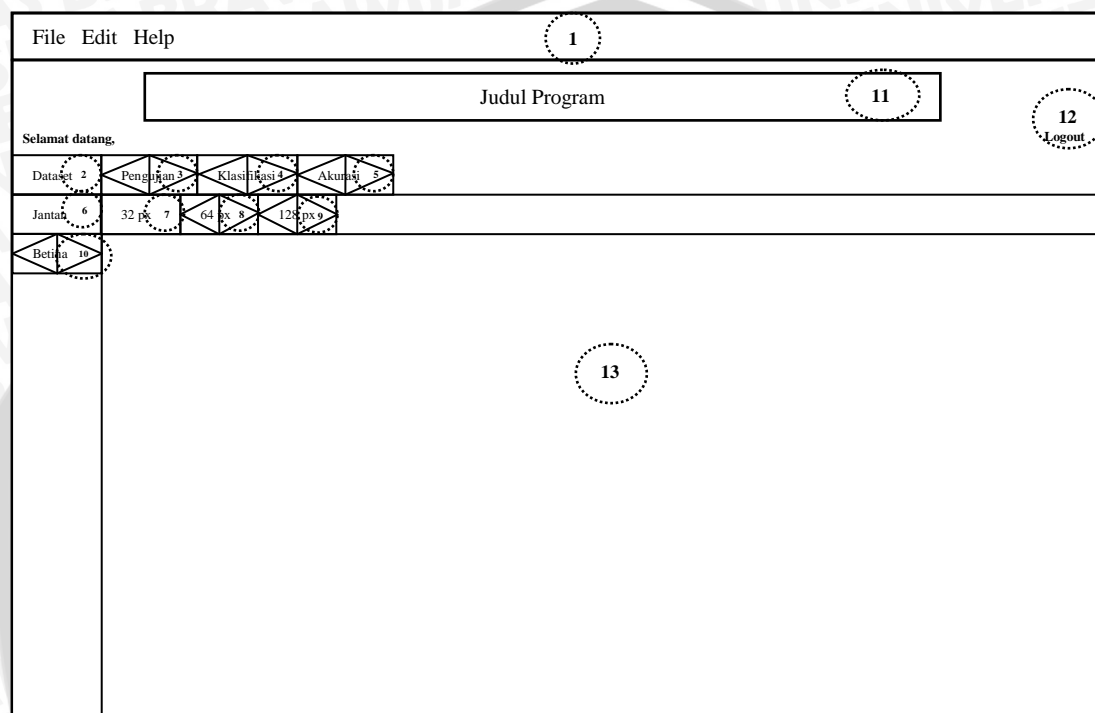
Keterangan gambar 4.69:

1. Menu bar
2. Tabpane untuk data Sapi Bali
3. Tabpane untuk olah data pengguna
4. Tabel untuk data pengguna
5. Tombol tambah untuk menambah data pengguna
6. Tombol ubah untuk mengubah data pengguna
7. Tombol hapus untuk menghapus data pengguna
8. Judul program
9. Tombol logout untuk kembali ke halaman login



#### 4.2.4.4 Antarmuka Peternak

Pada antarmuka peternak terdapat tabpane dataset yang digunakan untuk menampilkan dataset Sapi Bali, pengujian, digunakan untuk melakukan proses skenario pengujian, klasifikasi dan akurasi. Rancangan antarmuka peternak ditunjukkan pada gambar 4.70.



Gambar 4.70 Antarmuka Peternak  
Sumber: [Perancangan]

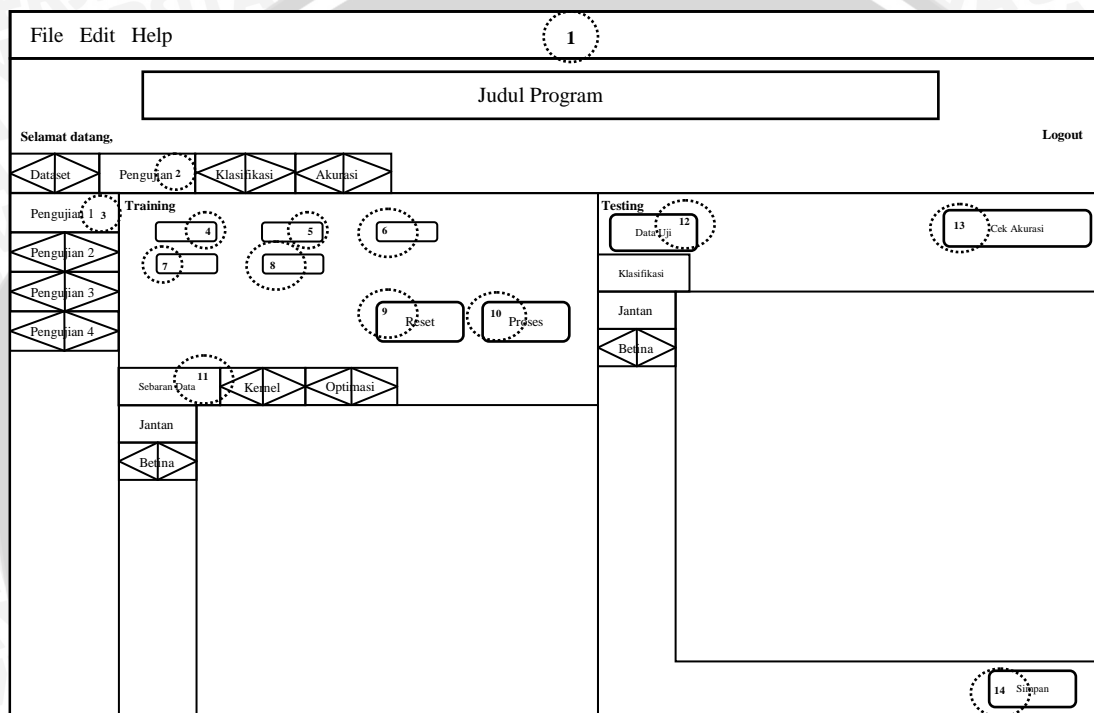
Keterangan gambar 4.70:

1. Menu bar
2. Tabpane untuk dataset Sapi Bali
3. Tabpane untuk skenario pengujian
4. Tabpane untuk klasifikasi data uji
5. Tabpane untuk akurasi skenario pengujian
6. Tabpane untuk Sapi Bali Jantan
7. Tabpane untuk data ukuran *cropping* 32 px
8. Tabpane untuk data ukuran *cropping* 64 px
9. Tabpane untuk data ukuran *cropping* 128 px
10. Tabpane untuk Sapi Bali Jantan

11. Judul program
12. Tombol logout untuk kembali ke halaman login
13. Tabel data ukuran *cropping* 32 px untuk Sapi Bali Jantan

#### 4.2.4.5 Antarmuka Pengujian

Pada antarmuka pengujian terdapat tabpane pengujian 1, pengujian 2, pengujian 3 dan pengujian 4. Rancangan antarmuka pengujian ditunjukkan pada gambar 4.71.



Gambar 4.71 Antarmuka Pengujian  
Sumber: [Perancangan]

Keterangan gambar 4.71:

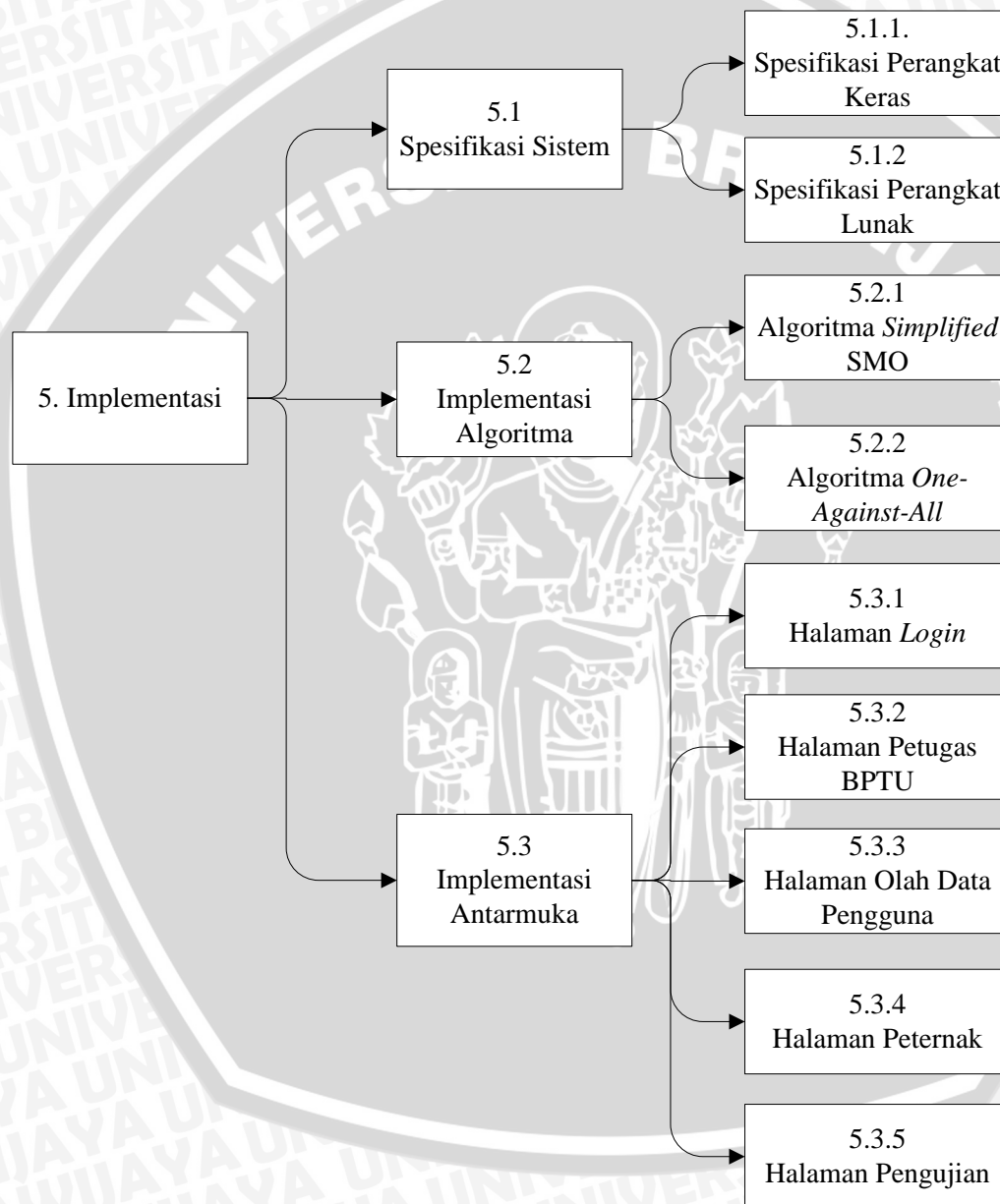
1. Menu bar
2. Tabpane untuk pengujian
3. Tabpane untuk pengujian 1 (rasio dan ukuran *cropping*)
4. Combobox rasio
5. Combobox iterasi
6. Combobox C
7. Combobox ukuran
8. Combobox gamma
9. Tombol reset
10. Tombol proses pelatihan

11. Tabpane sebaran data
12. Tombol load data uji
13. Tombol cek akurasi
14. Tombol simpan akurasi



## BAB V IMPLEMENTASI

Bab ini menjelaskan mengenai spesifikasi sistem, implementasi algoritma dan implementasi antarmuka sistem pendukung keputusan pemilihan bibit unggul Sapi Bali menggunakan algoritma *Simplified* SMO berdasarkan perancangan pada bab sebelumnya. Berikut pohon implementasi ditunjukkan pada gambar 5.1.



Gambar 5.1 Pohon Implementasi  
Sumber: [Implementasi]



## 5.1 Spesifikasi Sistem

Pada sub bab ini akan membahas spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam proses pengembangan sistem pendukung keputusan pemilihan bibit unggul Sapi Bali menggunakan algoritma *Simplified SMO*.

### 5.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pengembangan sistem pendukung keputusan dapat dilihat pada tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras

Perangkat Keras	Keterangan
Processor	Intel® Core™ i3 CPU M 380 @ 2.53 GHz (4 CPUs)
Memory RAM	2048 MB
Chip Type	ATI display adapter (0x68C1)
DAC Type	Internal DAC (400MHz)

Sumber : [Implementasi]

### 5.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem pendukung keputusan dapat dilihat pada tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Lunak

Perangkat Lunak	Keterangan
Sistem Operasi	Windows 7 Home Premium 32-bit
Bahas Pemrograman	Java
Tools	NetBeans IDE 7.1 RC2 dan Photoshop CS 6
Server	XAMPP 1.8.1
DBMS	MySQL

Sumber : [Implementasi]

## 5.2 Implementasi Algoritma

Sub bab implementasi algoritma akan menjelaskan tentang implementasi algoritma yang digunakan dalam pengembangan sistem pendukung keputusan pemilihan bibit unggul Sapi Bali. Algoritma yang akan dibahas dalam sub bab ini adalah algoritma *Simplified SMO* dan algoritma *One-Against-All*.

### 5.2.1 Algoritma *Simplified SMO*

Algoritma *Simplified SMO* merupakan algoritma yang digunakan untuk mencari nilai *alpha* dan *bias* optimal pada proses pelatihan data. Implementasi algoritma *Simplified SMO* dapat dilihat pada tabel 5.3.

Tabel 5.3 Implementasi Algoritma *Simplified SMO*

```

1 private void Simplified_SMO(Problem t, KernelParams p) {
2     int pass = 0;
3     int alpha_change = 0;
4     int i, j;
5     int as = 0;
6     model = new Model();
7     model.alpha = new double[t.l];
8     model.bias = 0;
9     model.params = p;
10    model.x = t.x;
11    model.y = t.y;
12    model.l = t.l;
13    model.n = t.n;
14    while (pass < maxpass) {
15        alpha_change = 0;
16        for (i = 0; i < t.l; i++) {
17            Ei = svmTest(t.x[i]) - t.y[i];
18            if ((t.y[i] * Ei < -tol && model.alpha[i] < C
19 || (t.y[i] * Ei > tol && model.alpha[i] > 0)) {
20                j = (int) Math.floor(Math.random() * (t.l - 1));
21                j = (j < i) ? j : (j + 1);
22                Ej = svmTest(t.x[j]) - t.y[j];
23                ai_lama = model.alpha[i];
24                aj_lama = model.alpha[j];
25                L = hitungL(t.y[i], t.y[j], i, j);
26                H = hitungH(t.y[i], t.y[j], i, j);
27                if (L == H) {
28                    continue;
29                }
30                double kij = kernel(model.x[i], model.x[j]);
31                double kii = kernel(model.x[i], model.x[i]);
32                double kjj = kernel(model.x[j], model.x[j]);
33                eta = 2 * kij - kii - kjj;
34                if (eta >= 0) {
35                    continue;
36                }
37                model.alpha[j] = model.alpha[j] - (t.y[j] *
38 (Ei - Ej)) / eta;
39                if (model.alpha[j] > H) {
40                    model.alpha[j] = H;
41                } else if (model.alpha[j] < L) {
42                    model.alpha[j] = L;
43                }
44                if (Math.abs(model.alpha[j] - aj_lama) <
45 tol2) {
46                    continue;
47                }
48                model.alpha[i] = model.alpha[i] + t.y[i] *
49 t.y[j] * (aj_lama - model.alpha[j]);
                    hitungBias(model.alpha[i], model.alpha[j], t.y[i],

```

```

50     t.y[j], kii, kjj, kij);
51         alpha_change++;
52     }
53 }
54 if (alpha_change == 0) {
55     pass++;
56 } else {
57     pass = 0;
58 }
59 }
60 System.out.println();

```

Sumber : [Implementasi]

- Baris 1 : sebuah metode bernama “Simplified\_SMO” yang memiliki parameter Problem dan KernelParams digunakan untuk mengambil fitur RGB dan kernel RBF.
- Baris 2-5 : untuk deklarasi variabel yang digunakan dalam algoritma *Simplified SMO*.
- Baris 14 : digunakan untuk mengecek batas maksimum perulangan dalam proses pelatihan.
- Baris 17 : digunakan untuk menghitung nilai error pada data ke-*i*.
- Baris 18-19 : digunakan untuk mengecek kondisi KKT pada setiap data ke-*i*.
- Baris 20-21 : digunakan untuk mendapatkan data pembandingan (data ke-*j*) secara acak (*Random*).
- Baris 22 : digunakan untuk menghitung nilai error pada data ke-*j*.
- Baris 23-24 : digunakan untuk menset nilai alpha awal kedalam variabel *ai\_lama* dan *aj\_lama*.
- Baris 25-29 : digunakan untuk menghitung nilai L dan H serta mengecek kesamaan nilai L dan H.
- Baris 33 : digunakan untuk menghitung nilai *eta*.
- Baris 36-43 : digunakan untuk menghitung nilai *alpha\_j* baru.
- Baris 47-48 : digunakan untuk menghitung nilai *alpha\_i* baru.
- Baris 49-50 : digunakan untuk menghitung nilai bias.
- Baris 51 : digunakan untuk menambah variabel  $alp = alp + 1$
- Baris 54-58 : digunakan untuk mengecek apakah ada perubahan pada nilai alpha dan *maxIterasi*

- Baris 1 : sebuah metode bernama “OneAgainstAll” yang memiliki parameter “FiturData” digunakan untuk mengambil fitur dari data uji
- Baris 2 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan pertama.
- Baris 3-5 : merupakan sebuah kondisi yang akan menghasikan klasifikasi “BURUK” apabila variabel hasil (pelatihan pertama) memberikan nilai  $< 0$ .
- Baris 6 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan kedua .
- Baris 7-11 : merupakan kondisi yang akan menghasilkan prediksi “BAIK” atau “SEDANG”.
- Baris 1 : sebuah metode bernama “OneAgainstAll” yang memiliki parameter “FiturData” digunakan untuk mengambil fitur dari data uji
- Baris 2 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan pertama.
- Baris 3-5 : merupakan sebuah kondisi yang akan menghasikan klasifikasi “BURUK” apabila variabel hasil (pelatihan pertama) memberikan nilai  $< 0$ .
- Baris 6 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan kedua .
- Baris 7-11 : merupakan kondisi yang akan menghasilkan prediksi “BAIK” atau “SEDANG”.
- Baris 1 : sebuah metode bernama “OneAgainstAll” yang memiliki parameter “FiturData” digunakan untuk mengambil fitur dari data uji
- Baris 2 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan pertama.
- Baris 3-5 : merupakan sebuah kondisi yang akan menghasikan klasifikasi “BURUK” apabila variabel hasil (pelatihan pertama) memberikan nilai  $< 0$ .
- Baris 6 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan kedua .





Baris 7-11 : merupakan kondisi yang akan menghasilkan prediksi “BAIK” atau “SEDANG”.

Baris 1 : sebuah metode bernama “OneAgainstAll” yang memiliki parameter “FiturData” digunakan untuk mengambil fitur dari data uji

Baris 2 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan pertama.

Baris 3-5 : merupakan sebuah kondisi yang akan menghasilkan klasifikasi “BURUK” apabila variabel hasil (pelatihan pertama) memberikan nilai  $< 0$ .

Baris 6 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan kedua .

Baris 7-11 : merupakan kondisi yang akan menghasilkan prediksi “BAIK” atau “SEDANG”.

Baris 1 : sebuah metode bernama “OneAgainstAll” yang memiliki parameter “FiturData” digunakan untuk mengambil fitur dari data uji

Baris 2 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan pertama.

Baris 3-5 : merupakan sebuah kondisi yang akan menghasilkan klasifikasi “BURUK” apabila variabel hasil (pelatihan pertama) memberikan nilai  $< 0$ .

Baris 6 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan kedua .

Baris 7-11 : merupakan kondisi yang akan menghasilkan prediksi “BAIK” atau “SEDANG”.

Baris 1 : sebuah metode bernama “OneAgainstAll” yang memiliki parameter “FiturData” digunakan untuk mengambil fitur dari data uji

Baris 2 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan pertama.

Baris 3-5 : merupakan sebuah kondisi yang akan menghasilkan klasifikasi “BURUK” apabila variabel hasil (pelatihan pertama) memberikan nilai  $< 0$ .

- Baris 6 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan kedua .
- Baris 7-11 : merupakan kondisi yang akan menghasilkan prediksi “BAIK” atau “SEDANG”.
- Baris 1 : sebuah metode bernama “OneAgainstAll” yang memiliki parameter “FiturData” digunakan untuk mengambil fitur dari data uji
- Baris 2 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan pertama.
- Baris 3-5 : merupakan sebuah kondisi yang akan menghasilkan klasifikasi “BURUK” apabila variabel hasil (pelatihan pertama) memberikan nilai  $< 0$ .
- Baris 6 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan kedua .
- Baris 7-11 : merupakan kondisi yang akan menghasilkan prediksi “BAIK” atau “SEDANG”.
- Baris 1 : sebuah metode bernama “OneAgainstAll” yang memiliki parameter “FiturData” digunakan untuk mengambil fitur dari data uji
- Baris 2 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan pertama.
- Baris 3-5 : merupakan sebuah kondisi yang akan menghasilkan klasifikasi “BURUK” apabila variabel hasil (pelatihan pertama) memberikan nilai  $< 0$ .
- Baris 6 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan kedua .
- Baris 7-11 : merupakan kondisi yang akan menghasilkan prediksi “BAIK” atau “SEDANG”.
- Baris 1 : sebuah metode bernama “OneAgainstAll” yang memiliki parameter “FiturData” digunakan untuk mengambil fitur dari data uji
- Baris 2 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan pertama.

### 5.2.2 Algoritma *One-Against-All*

Algoritma *One-Against-All* merupakan algoritma yang digunakan dalam proses klasifikasi. Implementasi algoritma *One-Against-All* dapat dilihat pada tabel 5.4.

Tabel 5.4 Implementasi Algoritma *One-Against-All*

```

1 private void OneAgainstAll(FiturData[] testing){
2     double hasil = Math.signum(svmJantan.svmTest(testing));
3     if (hasil < 0) {
4         pred = "BURUK";
5     } else {
6         hasil = Math.signum(svmxJantan.svmTest(testing));
7         if (hasil > 0) {
8             pred = "BAIK";
9         } else {
10            pred = "SEDANG";
11        }
12    }
}

```

Sumber : [Implementasi]

- Baris 1 : sebuah metode bernama “OneAgainstAll” yang memiliki parameter “FiturData” digunakan untuk mengambil fitur dari data uji
- Baris 2 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan pertama.
- Baris 3-5 : merupakan sebuah kondisi yang akan menghasilkan klasifikasi “BURUK” apabila variabel hasil (pelatihan pertama) memberikan nilai < 0.
- Baris 6 : digunakan untuk menghitung variabel hasil yang berisi fungsi keputusan pada pelatihan kedua .
- Baris 7-11 : merupakan kondisi yang akan menghasilkan prediksi “BAIK” atau “SEDANG”.

### 5.3 Implementasi Antarmuka

Implementasi antarmuka bertujuan untuk mengimplementasikan rancangan antarmuka pada sistem yang akan dibangun pada bab sebelumnya. Implementasi antarmuka akan menampilkan halaman login, halaman petugas BPTU, olah data pengguna, halaman peternak dan halaman pengujian.

#### 5.3.1 Halaman Login

Halaman login merupakan antarmuka yang tampil pada awal sistem berjalan. Pada halaman login terdapat *textfield* yang dapat digunakan untuk mengisi *username*

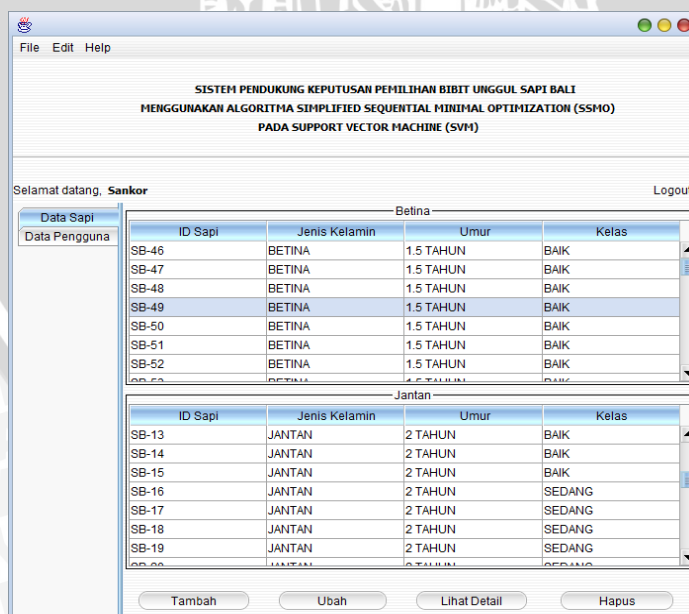
dan password oleh pengguna sistem. Implementasi halaman login dapat dilihat pada gambar 5.2.



Gambar 5.2 Halaman Login  
Sumber: [Implementasi]

### 5.3.2 Halaman Petugas BPTU

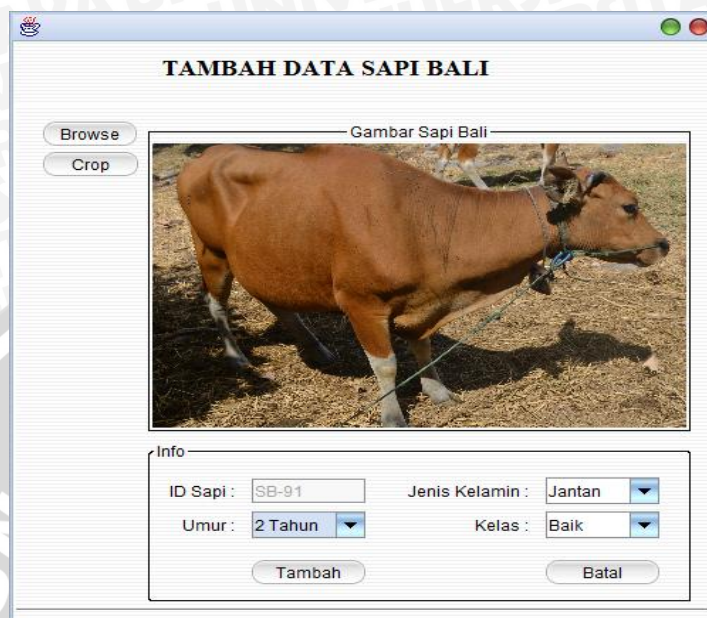
Tampilan awal halaman petugas BPTU menampilkan tabpane data Sapi Bali. Petugas BPTU dapat melakukan proses tambah, ubah, lihat detail dan hapus pada data Sapi Bali. Petugas BPTU dapat mengolah pengguna sistem yang terdapat pada tabpane data pengguna. Implementasi halaman petugas BPTU dapat dilihat pada gambar 5.3.



Gambar 5.3 Halaman Petugas BPTU  
Sumber: [Implementasi]



Halaman tambah data Sapi Bali terdapat sebuah tombol yang dapat digunakan untuk memilih dan menambahkan data Sapi Bali yang terdapat pada direktori komputer. Halaman tambah data Sapi Bali dapat dilihat pada gambar 5.4.



**TAMBAH DATA SAPI BALI**

Browse  
Crop

Gambar Sapi Bali

Info

ID Sapi : SB-91      Jenis Kelamin : Jantan  
Umur : 2 Tahun      Kelas : Baik

Tambah      Batal

Gambar 5.4 Halaman Tambah Data Sapi Bali  
Sumber: [Implementasi]

Pada halaman ubah data Sapi Bali petugas BPTU dapat mengubah informasi yang dimiliki pada data Sapi Bali. Halaman ubah data Sapi Bali dapat dilihat pada gambar 5.5.



**UBAH DATA SAPI BALI**

Ubah Gambar  
Ubah Crop

Citra Sapi Bali

Info

ID Sapi : SB-49  
Jenis Kelamin : Betina  
Umur : 1.5 Tahun  
Kelas : Baik

Simpan      Batal

Gambar 5.5 Halaman Ubah Data Sapi Bali  
Sumber: [Implementasi]

Pada halaman lihat detail data Sapi Bali terdapat informasi nilai rata-rata RGB setiap ukuran *cropping*. Halaman lihat detail data Sapi Bali dapat dilihat pada gambar 5.6.

**Data Sapi Bali**

Info Sapi Bali

ID Sapi : SB-49

Jenis Kelamin : BETINA

Umur : 1.5 TAHUN

Kelas : BAIK

Crop Sapi Bali

ID Crop	ID Sapi	Ukuran	Status	Red	Green
SC-32-49-0	SB-49	32px	Tidak Contr...	84.0	54.666666...
SC-64-49-0	SB-49	64px	Tidak Contr...	84.666666...	54.333333...
SC-128-49-0	SB-49	128px	Tidak Contr...	85.666666...	55.666666...

Lihat Data    Kembali

Gambar 5.6 Halaman Lihat Detail Data Sapi Bali  
Sumber: [Implementasi]

### 5.3.3 Halaman Olah Data Pengguna

Pada halaman olah data pengguna petugas BPTU terdapat tabel yang berisi informasi pengguna sistem. Petugas BPTU dapat melakukan penambahan, pengubahan hak akses dan penghapusan pada setiap pengguna sistem. Implementasi halaman olah data pengguna dapat dilihat pada gambar 5.7.

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

Selamat datang, Sankor    Logout

ID Pengguna	Nama	Alamat	Jenis Kelamin	Password	Status
10503	Sankor	Barat Padepok...	L	186b690e2989...	PTGS BPTU
11111	Korsan	Malang Aja	L	b0baee9d279d...	PTGS BPTU
12311	killua	gunung Zoldyck	L	35e2dcdba19...	PTK
22222	Yosep	Padepokan	L	3d2172418ce3...	PTK

Tambah    Ubah    Hapus

Gambar 5.7 Halaman Olah Data Pengguna  
Sumber: [Implementasi]

### 5.3.4 Halaman Peternak

Pada tampilan halaman peternak terdapat beberapa tabpane utama yaitu dataset, pengujian, klasifikasi dan akurasi. Tabpane dataset berisi data-data Sapi Bali yang terdapat pada *database*. Tabpane pengujian digunakan untuk melakukan skenario pengujian yang telah ditetapkan sebelumnya. Tabpane klasifikasi digunakan untuk melakukan menguji sebuah data uji. Tabpane akurasi digunakan untuk menyimpan tingkat akurasi yang dihasilkan pada setiap skenario pengujian. Halaman peternak dapat dilihat pada gambar 5.8.

ID	Red	Green	Blue	Kelas
SB-13	19.333333333333332	29.0	36.333333333333336	BAIK
SB-3	122.0	119.66666666666667	124.0	BAIK
SB-6	83.33333333333333	71.66666666666667	79.0	BAIK
SB-11	78.66666666666667	80.0	95.0	BAIK
SB-9	75.33333333333333	73.66666666666667	72.33333333333333	BAIK
SB-14	90.0	83.33333333333333	106.66666666666667	BAIK
SB-1	126.0	126.0	133.66666666666666	BAIK
SB-5	122.33333333333333	127.0	157.66666666666666	BAIK
SB-7	115.66666666666667	106.0	114.0	BAIK
SB-12	65.0	62.333333333333336	65.0	BAIK
SB-15	18.666666666666668	27.666666666666668	34.666666666666664	BAIK
SB-2	130.66666666666666	126.66666666666667	131.33333333333334	BAIK
SB-4	116.66666666666667	113.0	117.66666666666667	BAIK
SB-10	127.66666666666667	123.0	127.66666666666667	BAIK
SB-8	126.0	127.0	143.33333333333334	BAIK
SB-25	121.33333333333333	97.33333333333333	79.33333333333333	SEDANG
SB-27	123.66666666666667	97.0	77.66666666666667	SEDANG
SB-16	108.66666666666667	82.33333333333333	65.0	SEDANG
SB-30	95.66666666666667	87.0	84.66666666666667	SEDANG
SB-19	115.66666666666667	94.66666666666667	82.66666666666667	SEDANG

Gambar 5.8 Halaman Peternak  
Sumber: [Implementasi]

### 5.3.5 Halaman Pengujian

Pada halaman pengujian peternak dapat melakukan skenario pengujian dengan mengubah nilai parameter yang akan digunakan untuk proses pelatihan data. Halaman pengujian dapat dilihat pada gambar 5.9.

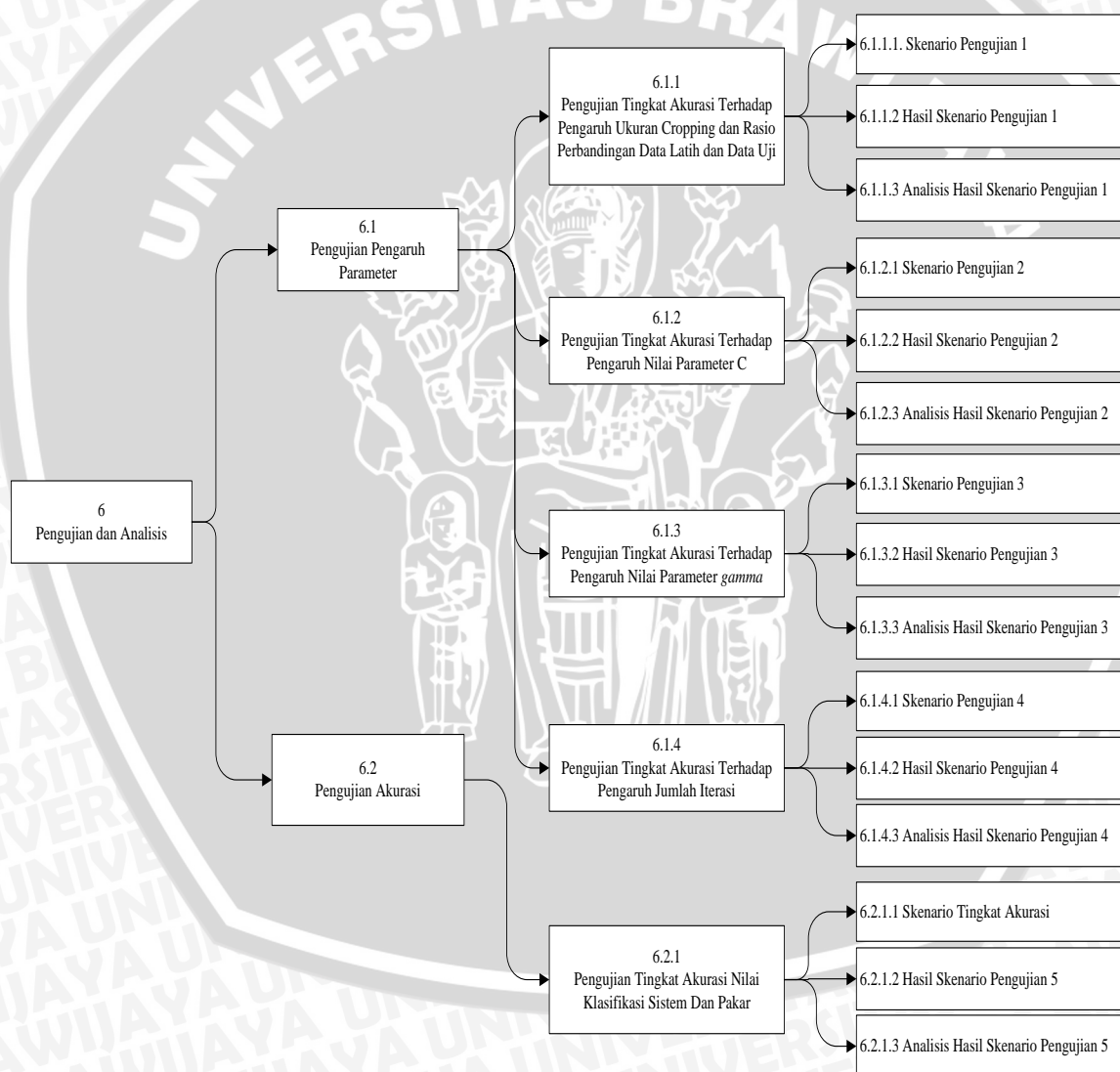
Citra ke-1	Citra ke-2	Citra ke-3	Citra ke-4	Citra ke-5	Citra ke-6	Citra ke-7	Citra ke-8	Citra ke-9	Citra ke-10	Citra ke-11	Citra ke-12	Citra ke-13	Citra ke-14
SB-31	224.0	165.0	111.66666666666666	BURUK	BURUK								
SB-9	75.33333333333333	73.66666666666667	72.33333333333333	BAIK	SEDANG								
SB-28	125.0	84.0	72.33333333333333	SEDANG	BURUK								
SB-27	123.66666666666667	97.0	77.66666666666667	SEDANG	BURUK								
SB-40	160.33333333333333	113.66666666666666	78.33333333333333	BURUK	BURUK								
SB-1	126.0	128.0	133.66666666666666	BAIK	BURUK								
SB-28	117.66666666666666	91.66666666666666	73.66666666666666	SEDANG	BURUK								
SB-21	138.66666666666666	132.0	130.0	SEDANG	BURUK								
SB-25	121.33333333333333	97.33333333333333	79.33333333333333	SEDANG	BURUK								
SB-11	78.66666666666666	80.0	95.0	BAIK	BAIK								
SB-42	188.66666666666666	168.66666666666666	148.0	BURUK	BURUK								
SB-8	126.0	127.0	143.33333333333333	BAIK	BURUK								
SB-12	65.0	62.33333333333333	65.0	BAIK	BAIK								
SB-44	254.0	212.33333333333333	163.33333333333333	BURUK	BURUK								
SB-41	155.66666666666666	107.33333333333333	69.0	BURUK	BURUK								

Gambar 5.9 Halaman Pengujian  
Sumber: [Implementasi]

## BAB VI

### PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai pengujian dan analisis sistem pendukung keputusan pemilihan bibit unggul Sapi Bali dengan menggunakan algoritma *Simplified SMO* pada *Support Vector Machine* (SVM). Proses pengujian pada penelitian ini dilakukan dengan pengujian akurasi terhadap skenario yang telah ditentukan sebelumnya. Analisis hasil pengujian dilakukan untuk menganalisa hasil pengujian yang telah dilakukan. Gambar 6.1 merupakan pohon pengujian dan analisis.



Gambar 6.1 Pohon Pengujian dan Analisis  
 Sumber : [Pengujian dan Analisis]





## 6.1 Pengujian Pengaruh Parameter

Pada sub bab ini akan menjelaskan pengaruh parameter yang terdapat pada penggunaan algoritma *Simplified SMO* terhadap tingkat akurasi yang dihasilkan. Parameter yang digunakan antara lain nilai  $C$ , nilai *gamma* ( $\gamma$ ), dan jumlah iterasi.

### 6.1.1 Pengujian Tingkat Akurasi Terhadap Pengaruh Ukuran *Cropping* dan Rasio Perbandingan Data Latih dan Data Uji

Pada sub bab ini akan menjelaskan tentang pengujian tingkat akurasi terhadap pengaruh ukuran *cropping* dan rasio perbandingan data latih dan data uji, hasil skenario serta analisis hasil dari skenario pengujian pertama.

#### 6.1.1.1 Skenario Pengujian 1

Pada skenario pengujian 1 bertujuan untuk mendapatkan rata-rata tingkat akurasi terbaik dari lima kali percobaan pada masing-masing rasio dan ukuran *cropping* yang akan digunakan untuk proses skenario pengujian 1. Data latih dan data uji yang digunakan diambil secara acak (*random*) dari keseluruhan dataset pada masing-masing jenis kelamin dan kelas pada Sapi Bali. Pengujian ini dilakukan dengan menggunakan jumlah iterasi = 100,  $C = 16$ , *gamma* = 0.01. Jumlah data latih dan data uji pada penelitian ini dapat dilihat pada tabel 6.1.

Tabel 6.1 Perbandingan Jumlah Data Latih dan Data Uji

Rasio	Data Latih	Data Uji
90:10	39	6
80:20	36	9
70:30	30	15
60:40	27	18
50:50	21	24

Sumber : [Pengujian dan Analisis]

#### 6.1.1.2 Hasil Skenario Pengujian 1

Nilai parameter yang digunakan secara konstan pada pengujian ini yaitu  $C = 16$ , *gamma* ( $\gamma$ ) = 0.01 dan jumlah iterasi = 100. Berikut ini adalah hasil pengujian tingkat akurasi berdasarkan pengaruh ukuran *cropping* dan rasio perbandingan data latih dan data uji.

##### A. Sapi Bali Jantan

##### 1. Akurasi Sapi Bali jantan dengan ukuran *cropping* 32 px

Tabel 6.2 Tingkat Akurasi Sapi Bali Jantan dengan Ukuran *Cropping* 32 px (%)

Rasio	Hasil Percobaan ke- (%)					Rata-rata(%)
	1	2	3	4	5	
90:10	50	83.33	66.67	50	50	60

80:20	66.67	66.67	88.89	66.67	66.67	71.11
70:30	60	73.33	60	53.33	53.33	60
60:40	66.67	77.78	66.67	72.22	50	66.67
50:50	50	58.33	70.83	66.67	70.83	63.33

Sumber : [Pengujian]

2. Akurasi Sapi Bali jantan dengan ukuran *cropping* 64 px

Tabel 6.3 Tingkat Akurasi Sapi Bali Jantan dengan Ukuran *Cropping* 64 px (%)

Data Latih	Hasil Percobaan ke- (%)					Rata-rata(%)
	1	2	3	4	5	
90:10	50	100	50	83.33	50	66.67
80:20	55.56	88.89	55.56	88.89	66.67	71.11
70:30	80	80	66.67	60	53.33	68
60:40	55.56	83.33	61.11	83.33	61.11	68.89
50:50	58.33	45.83	75	62.5	62.5	60.83

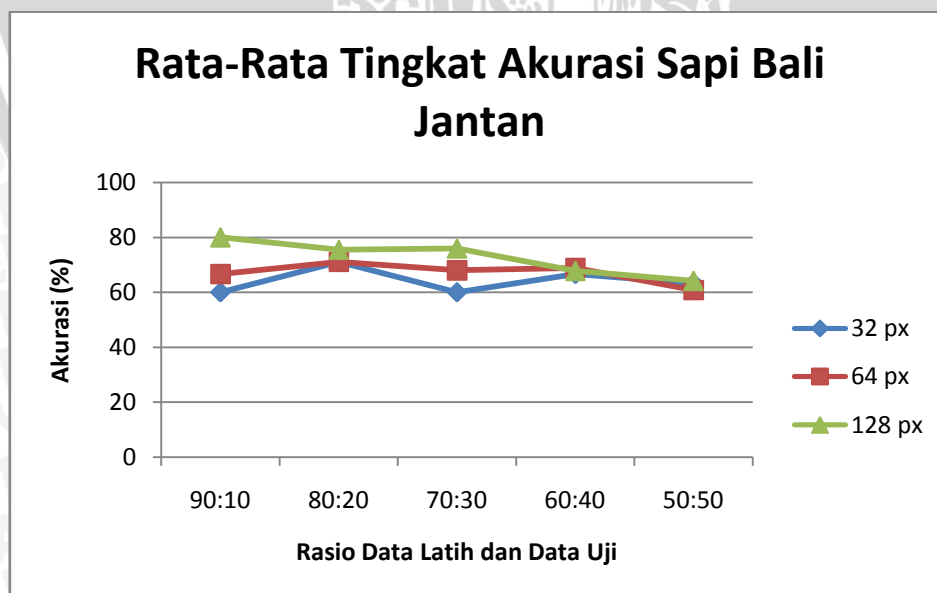
Sumber : [Pengujian]

3. Akurasi Sapi Bali jantan dengan ukuran *cropping* 128 px

Tabel 6.4 Tingkat Akurasi Sapi Bali Jantan dengan Ukuran *Cropping* 128px (%)

Data Latih	Hasil Percobaan ke- (%)					Rata-rata(%)
	1	2	3	4	5	
90:10	83.33	83.33	83.33	66.67	83.33	80
80:20	88.89	55.56	88.89	55.56	88.89	75.56
70:30	73.33	86.67	73.33	66.67	80	76
60:40	77.78	66.67	55.56	61.11	77.78	67.78
50:50	62.5	70.83	62.5	50	75	64.167

Sumber : [Pengujian]



Gambar 6.2 Grafik Tingkat Akurasi Sapi Bali Jantan Terhadap Ukuran *Cropping* dan Rasio Perbandingan Data Laith dan data Uji

Sumber : [Pengujian]

Berdasarkan hasil pengujian terhadap ukuran *cropping* dan perbandingan rasio data latih dan data uji pada Sapi Bali Jantan didapatkan rata-rata tingkat akurasi tertinggi sebesar 80 %. 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.

## B. Sapi Bali Betina

### 1. Akurasi Sapi Bali betina dengan ukuran *cropping* 32 px

Tabel 6.5 Tingkat Akurasi Sapi Bali Betina dengan Ukuran *Cropping* 32 px (%)

Rasio	Hasil Percobaan ke- (%)					Rata-rata(%)
	1	2	3	4	5	
90:10	50	50	66.67	33.33	50	50
80:20	77.78	33.33	44.44	44.44	33.33	46.67
70:30	46.67	53.33	53.33	60	53.33	53.33
60:40	44.44	55.56	50	50	55.56	51.11
50:50	45.83	45.83	58.33	45.83	37.5	46.67

Sumber : [Pengujian]

### 2. Akurasi Sapi Bali betina dengan ukuran *cropping* 64 px

Tabel 6.6 Tingkat Akurasi Sapi Bali Betina dengan Ukuran *Cropping* 64 px (%)

Rasio	Hasil Percobaan ke- (%)					Rata-rata(%)
	1	2	3	4	5	
90:10	50	50	50	50	50	50
80:20	88.89	55.56	44.44	66.67	33.33	57.78
70:30	66.67	46.67	53.33	40	46.67	50.67
60:40	33.33	55.56	50	55.56	38.89	46.67
50:50	45.83	50	37.5	37.5	37.5	41.67

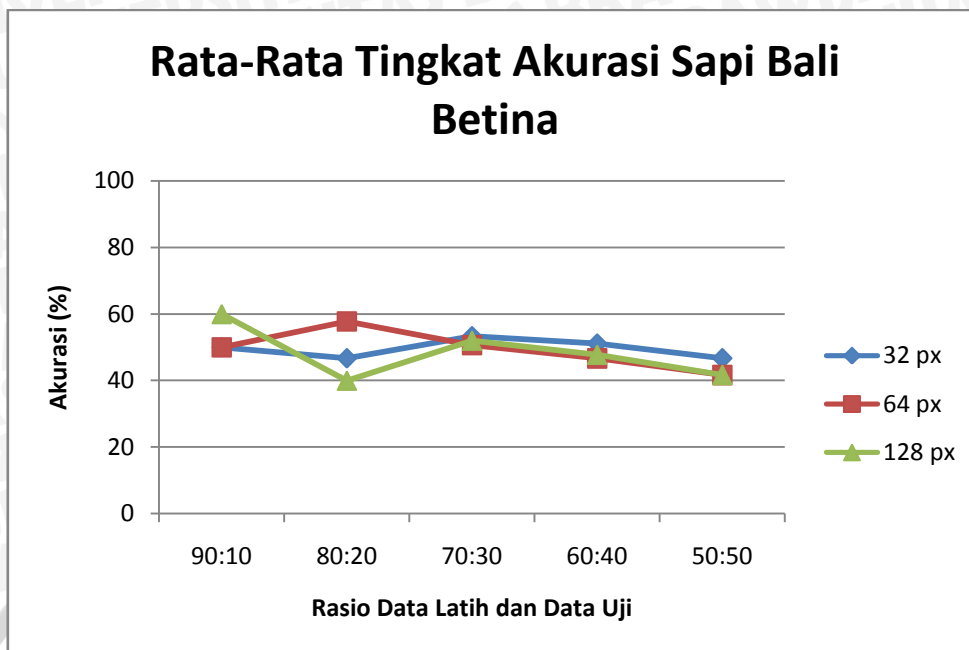
Sumber : [Pengujian]

### 3. Akurasi Sapi Bali betina dengan ukuran *cropping* 128 px

Tabel 6.7 Tingkat Akurasi Sapi Bali Betina dengan Ukuran *Cropping* 128 px (%)

Rasio	Hasil Percobaan ke- (%)					Rata-rata(%)
	1	2	3	4	5	
90:10	66.67	66.67	66.67	66.67	33.33	60
80:20	33.33	22.22	55.56	44.44	44.44	40
70:30	53.33	40	33.33	60	73.33	52
60:40	55.56	55.56	38.89	55.56	33.33	47.78
50:50	37.5	58.33	25	41.67	45.83	41.67

Sumber : [Pengujian]



Gambar 6.3 Grafik Tingkat Akurasi Sapi Bali Betina Terhadap Ukuran *Cropping* dan Perbandingan Rasio  
Sumber : [Pengujian]

Berdasarkan hasil pengujian terhadap ukuran *cropping* dan perbandingan rasio data latih dan data uji pada Sapi Bali Betina didapatkan rata-rata tingkat akurasi tertinggi sebesar 60 %. Tingkat akurasi tersebut terdapat pada perbandingan rasio 90:10 dan ukuran *cropping* 128 px. Dengan demikian ukuran *cropping* 128 px dan rasio perbandingan 90:10 untuk data latih dan data uji dan akan digunakan untuk skenario pengujian kedua pada Sapi Bali Betina.

#### 6.1.1.3 Analisis Hasil Skenario Pengujian 1

Hasil 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 80 % dan 60 %. Pada Sapi Bali jantan betina ukuran *cropping* dan rasio perbandingan data latih dan data uji yaitu 128 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.

#### 6.1.2 Pengujian Tingkat Akurasi Terhadap Pengaruh Nilai Parameter C

Pada sub bab ini akan menjelaskan tentang skenario pengujian kedua yaitu pengujian tingkat akurasi terhadap pengaruh nilai parameter C pada algoritma *Simplified SMO*, hasil skenario serta analisis hasil dari skenario pengujian kedua.

### 6.1.2.1 Skenario Pengujian 2

Pada skenario pengujian 2 bertujuan untuk mendapatkan rata-rata tingkat akurasi terbaik dari lima kali percobaan pada masing-masing nilai parameter. Pengujian ini dilakukan dengan menaikkan dan menurunkan nilai parameter C yang pada awal pengujian nilai C yang digunakan yaitu 16. Nilai parameter C akan dinaikkan dari 16 menjadi 64 dan 256 sedangkan nilai parameter yang akan diturunkan yaitu dari 16 menjadi 4 dan 1. Parameter lain yang digunakan dalam pengujian ini adalah menggunakan jumlah iterasi = 100,  $\gamma$  0.01 dan ukuran serta rasio data latih dan data uji yang didapatkan dari tingkat akurasi terbaik dari skenario pengujian 1.

### 6.1.2.2 Hasil Skenario Pengujian 2

Pengujian ini menggunakan rasio dan ukuran *cropping* yang telah didapatkan pada skenario pengujian pertama serta nilai parameter  $\gamma$  = 0.01 dan jumlah iterasi = 100. Pada Sapi Bali jantan dan betina rasio perbandingan dan ukuran *cropping* yang terbaik adalah 90:10 dan 128 px. Berikut ini adalah hasil pengujian tingkat akurasi berdasarkan pengaruh nilai parameter C.

#### A. Sapi Bali Jantan

- Akurasi Sapi Bali jantan dengan ukuran *cropping* 128 px dan rasio 90:10

Tabel 6.8 Tingkat Akurasi Sapi Bali Jantan Terhadap Pengaruh Nilai C (%)

C	Hasil Percobaan ke- (%)					Rata-rata(%)
	1	2	3	4	5	
1	100	83.33	100	50	66.67	80
4	66.67	83.33	66.67	66.67	100	76.67
16	66.67	83.33	100	66.67	83.33	80
64	100	83.33	66.67	83.33	100	86.67
256	83.33	66.67	100	66.67	100	83.33

Sumber : [Pengujian]

#### B. Sapi Bali Betina

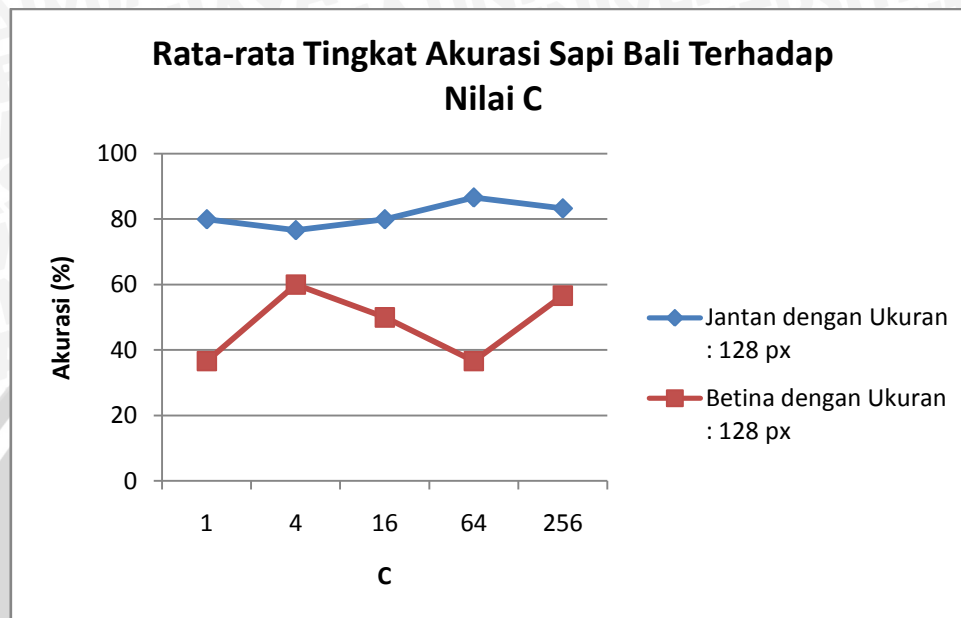
- Akurasi Sapi Bali betina dengan ukuran *cropping* 128 px dan rasio 90:10

Tabel 6.9 Tingkat Akurasi Sapi Bali Betina Terhadap Pengaruh Nilai C (%)

C	Hasil Percobaan ke- (%)					Rata-rata(%)
	1	2	3	4	5	
1	33.33	50	16.67	33.33	50	36.67
4	66.67	33.33	66.67	66.67	66.67	60

16	83.33	33.33	50	50	33.33	50
64	66.67	33.33	33.33	16.67	33.33	36.67
256	50	50	66.67	50	66.67	56.67

Sumber : [Pengujian]



Gambar 6.4 Grafik Tingkat Akurasi Sapi Bali Terhadap Nilai Parameter C

Sumber : [Pengujian]

Berdasarkan hasil pengujian terhadap nilai C didapatkan rata-rata tingkat akurasi tertinggi sebesar 86.67 % pada Sapi Bali jantan dengan nilai C = 64 dan tingkat akurasi tertinggi pada Sapi Bali betina sebesar 60 % dengan nilai C = 4. Dengan demikian tingkat akurasi terbaik pada Sapi Bali jantan dan betina akan di gunakan untuk pengujian ketiga.

### 6.1.2.3 Analisis Skenario Pengujian 2

Pengujian kedua Sapi Bali jantan dan Sapi Bali betina untuk tingkat akurasi terhadap pengaruh nilai parameter C memiliki tingkat akurasi tertinggi sebesar 86.67% dan 60%. Pada Sapi Bali jantan nilai parameter C = 64 merupakan nilai parameter yang terbaik pada Sapi Bali jantan dan nilai parameter C = 4 merupakan nilai parameter terbaik pada Sapi Bali betina. Berdasarkan tabel 6.8 pada Sapi Bali jantan dapat dilihat bahwa ketika nilai parameter C dinaikkan sebanyak 2 nilai dari nilai C = 16 terjadi peningkatan rata-rata tingkat akurasi sebesar 86.67% pada nilai C = 64 dan 83.33% pada nilai C = 256. Pada nilai parameter C yang diturunkan didapatkan penurunan rata-rata tingkat akurasi sebesar 76.67% pada nilai C = 4 dan 60% pada nilai C = 1. Berdasarkan tabel 6.9 pada Sapi Bali betina dapat dilihat

bahwa ketika nilai parameter  $C$  dinaikkan sebanyak 2 tahap dari nilai  $C = 16$  terjadi peningkatan rata-rata tingkat akurasi. Pada nilai parameter  $C = 64$  dan  $256$  memiliki rata-rata tingkat akurasi sebesar  $36.67\%$  dan  $56.67\%$ . Pada nilai parameter  $C$  yang diturunkan didapatkan rata-rata tingkat akurasi sebesar  $60\%$  dan  $36.67\%$ .

Hal ini dapat disimpulkan bahwa pengaruh nilai parameter  $C$  pada Sapi Bali jantan akan meningkatkan hasil rata-rata tingkat akurasi apabila nilai parameter  $C$  dinaikkan sedangkan jika nilai parameter  $C$  diturunkan akan menurunkan hasil rata-rata tingkat akurasi. Pada Sapi Bali betina terjadi inkonsistenan dari rata-rata tingkat akurasi yang dihasilkan ketika nilai parameter  $C$  dinaikkan atau diturunkan. Hal ini dikarenakan penggunaan nilai  $C$  pada algoritma *Simplified SMO* akan mempengaruhi klasifikasi data latih pada proses pelatihan. Jadi, semakin kecil nilai  $C$  akan mengakibatkan semakin sedikit fungsi *hyperplane* yang konsisten pada proses pelatihan.

### 6.1.3 Pengujian Tingkat Akurasi Terhadap Pengaruh Nilai Parameter $\Gamma$ ( $\gamma$ )

Pada sub bab ini akan menjelaskan tentang skenario pengujian ketiga yaitu pengujian tingkat akurasi terhadap pengaruh nilai parameter  $\gamma$  pada kernel *Radial Basis Function* (RBF), hasil skenario serta analisis hasil dari skenario pengujian ketiga.

#### 6.1.3.1 Skenario Pengujian 3

Pada pengujian skenario 3 bertujuan untuk mendapatkan rata-rata tingkat akurasi terbaik dari lima kali percobaan pada masing-masing nilai parameter  $\gamma$ . Pengujian ini dilakukan dengan menaikkan dan menurunkan nilai parameter  $\gamma$  yang pada awal pengujian nilai  $\gamma$  yang digunakan yaitu  $0.01$ . Nilai parameter  $\gamma$  akan dinaikkan dari  $0.01$  menjadi  $0.1$  dan  $1$  sedangkan nilai parameter yang akan diturunkan yaitu dari  $0.01$  menjadi  $0.001$  dan  $0.0001$ . Parameter lain yang digunakan dalam pengujian ini adalah menggunakan jumlah iterasi =  $100$ , nilai  $C$  dan ukuran serta rasio data latih dan data uji yang didapatkan dari tingkat akurasi terbaik dari skenario pengujian 1 dan 2.

#### 6.1.3.2 Hasil Skenario Pengujian 3

Pengujian ini menggunakan rasio, ukuran *cropping* dan nilai parameter  $C$  yang telah didapatkan pada skenario pengujian sebelumnya serta jumlah terasi =  $100$ . Pada Sapi Bali jantan rasio perbandingan, ukuran *cropping* dan nilai  $C$  yang terbaik

adalah 90:10, 128 px dan 64 sedangkan pada Sapi Bali betina rasio perbandingan, ukuran *cropping* dan nilai C yang terbaik adalah 90:10, 128 px dan 4. Berikut ini adalah hasil pengujian tingkat akurasi berdasarkan pengaruh nilai parameter *gamma*.

#### A. Sapi Bali Jantan

- Akurasi Sapi Bali jantan dengan ukuran *cropping* 128 px, rasio 90:10 dan C = 64

Tabel 6.10 Tingkat Akurasi Akurasi Sapi Bali Jantan Terhadap Pengaruh Nilai *Gamma* ( $\gamma$ ) (%)

<i>gamma</i> ( $\gamma$ )	Hasil Percobaan ke- (%)					Rata-rata(%)
	1	2	3	4	5	
0.0001	100	100	100	83.33	100	96.67
0.001	100	66.67	66.67	100	83.33	83.33
0.01	66.67	83.33	83.33	100	83.33	83.33
0.1	33.33	50	33.33	50	33.33	40
1	33.33	33.33	33.33	33.33	33.33	33.33

Sumber : [Pengujian]

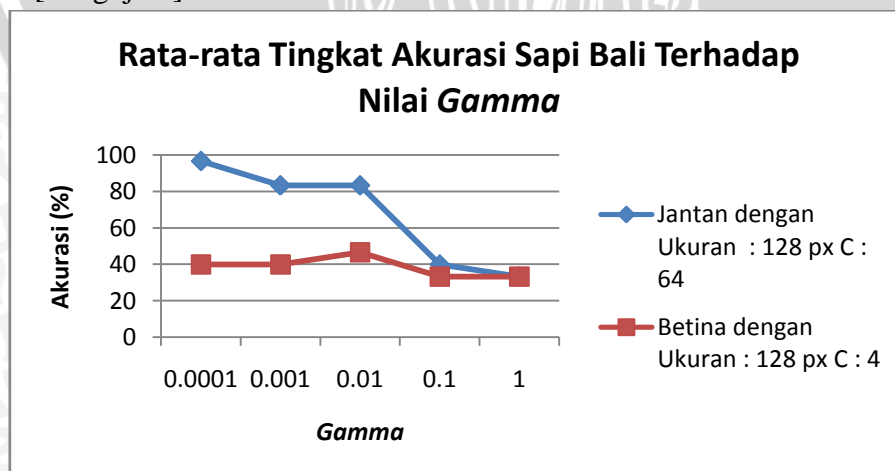
#### B. Sapi Bali Betina

- Akurasi Sapi Bali betina dengan ukuran *cropping* 64 px, rasio 90:10 dan C = 4

Tabel 6.11 Tingkat Akurasi Akurasi Sapi Bali Betina Terhadap Pengaruh Nilai *Gamma* ( $\gamma$ ) (%)

<i>gamma</i> ( $\gamma$ )	Percobaan ke-					Rata-rata
	1	2	3	4	5	
0.0001	16.67	66.67	33.33	50	33.33	40
0.001	50	16.67	33.33	66.67	33.33	40
0.01	50	50	50	50	33.33	46.67
0.1	33.33	33.33	33.33	33.33	33.33	33.33
1	33.33	33.33	33.33	33.33	33.33	33.33

Sumber : [Pengujian]



Gambar 6.5 Grafik Tingkat Akurasi Sapi Bali Terhadap Nilai Parameter *Gamma* ( $\gamma$ )

Sumber : [Pengujian]



Berdasarkan hasil pengujian terhadap nilai  $\gamma$  didapatkan rata-rata tingkat akurasi tertinggi sebesar 96.67 % pada Sapi Bali jantan dengan nilai  $\gamma$  ( $\gamma$ ) = 0.0001. Pada pengujian Sapi Bali betina terdapat dua buah tingkat akurasi terbaik yang sama yaitu pada nilai  $\gamma$  ( $\gamma$ ) 0.01 dengan tingkat akurasi sebesar 46.67%.

### 6.1.3.3 Analisis Skenario Pengujian 3

Pada pengujian ketiga Sapi Bali jantan dan Sapi Bali betina untuk tingkat akurasi terhadap pengaruh nilai parameter  $\gamma$  ( $\gamma$ ) memiliki tingkat akurasi tertinggi sebesar 96.67% dan 46.67%. Berdasarkan tabel 6.10 pada Sapi Bali jantan dapat dilihat bahwa ketika nilai  $\gamma$  ( $\gamma$ ) dinaikkan sebanyak 2 nilai dari nilai  $\gamma$  ( $\gamma$ ) = 0.01 terjadi penurunan rata-rata tingkat akurasi sebesar 40% dan 33.33% sedangkan ketika nilai  $\gamma$  ( $\gamma$ ) diturunkan terjadi peningkatan rata-rata tingkat akurasi sebesar 83.33% pada nilai  $\gamma$  ( $\gamma$ ) = 0.001 dan 96.67% nilai  $\gamma$  ( $\gamma$ ) = 0.0001. Berdasarkan tabel 6.11 pada Sapi Bali betina dapat dilihat bahwa ketika nilai  $\gamma$  ( $\gamma$ ) dinaikkan ataupun diturunkan terjadi penurunan tingkat akurasi dari nilai  $\gamma$  ( $\gamma$ ) = 0.01 sebesar 40% untuk masing-masing nilai  $\gamma$  ( $\gamma$ ) = 0.001, 0.0001 dan 33.33% untuk masing-masing  $\gamma$  ( $\gamma$ ) = 0.1, 1.

Hal ini dapat disimpulkan bahwa pengaruh nilai parameter  $\gamma$  ( $\gamma$ ) pada Sapi Bali jantan dan betina akan menghasilkan rata-rata tingkat akurasi lebih tinggi jika nilai  $\gamma$  ( $\gamma$ ) diturunkan daripada nilai  $\gamma$  ( $\gamma$ ) dinaikkan. Nilai parameter  $\gamma$  ( $\gamma$ ) pada fungsi kernel *Radial Basis Function* (RBF) digunakan untuk menentukan kedekatan antar dua titik data latih. Jadi semakin kecil nilai  $\gamma$  ( $\gamma$ ) maka kedekatan antar dua titik data akan semakin besar sehingga memudahkan untuk menemukan garis pemisah (*hyperplane*) terbaik antar dua kelas. Hal ini dapat dilihat dari hasil tingkat akurasi pada nilai  $\gamma$  ( $\gamma$ ) < 0.01 lebih baik dari pada nilai  $\gamma$  ( $\gamma$ ) > 0.01.

### 6.1.4 Pengujian Tingkat Akurasi Terhadap Pengaruh Jumlah Iterasi

Pada sub bab ini akan menjelaskan tentang skenario pengujian keempat yaitu pengujian tingkat akurasi terhadap pengaruh jumlah iterasi, hasil skenario serta analisis hasil dari skenario pengujian keempat.

#### 6.1.4.1 Skenario Pengujian 4

Pada skenario pengujian 4 bertujuan untuk mendapatkan rata-rata tingkat akurasi terbaik dari lima kali percobaan pada masing-masing jumlah iterasi. Pengujian ini dilakukan dengan menaikkan dan menurunkan jumlah iterasi yang pada awal pengujian jumlah iterasi yang digunakan yaitu 100. Jumlah iterasi akan dinaikkan dari 100 menjadi 1000 dan 10000 sedangkan jumlah iterasi yang akan diturunkan yaitu dari 100 menjadi 10 dan 1.

#### 6.1.4.2 Hasil Skenario Pengujian 4

Pengujian ini menggunakan rasio, ukuran *cropping*, nilai parameter C dan nilai parameter *gamma* ( $\gamma$ ) terbaik berdasarkan tingkat akurasi yang telah didapatkan pada skenario pengujian sebelumnya. Pada Sapi Bali jantan rasio perbandingan, ukuran *cropping*, nilai C dan nilai *gamma* ( $\gamma$ ) yang terbaik adalah 90:10, 128 px, 256 dan 0.0001 sedangkan pada Sapi Bali betina rasio perbandingan, ukuran *cropping*, nilai C dan nilai *gamma* ( $\gamma$ ) yang terbaik adalah 90:10, 64 px, 256 dan 0.0001. Berikut ini adalah hasil pengujian tingkat akurasi berdasarkan pengaruh jumlah iterasi.

##### A. Sapi Bali Jantan

- Akurasi Sapi Bali jantan dengan ukuran *cropping* 128 px, rasio 90:10, C = 64 dan *gamma* ( $\gamma$ ) = 0.0001

Tabel 6.12 Tingkat Akurasi Sapi Bali Jantan Terhadap Pengaruh Jumlah Iterasi (%)

Iterasi	Hasil Percobaan ke- (%)					Rata-rata(%)
	1	2	3	4	5	
1	100	100	83.33	100	100	96.67
10	100	100	100	100	100	100
100	100	83.33	100	100	100	96.67
1000	100	100	100	100	100	100
10000	100	100	100	100	100	100

Sumber : [Pengujian]

##### B. Sapi Bali Betina

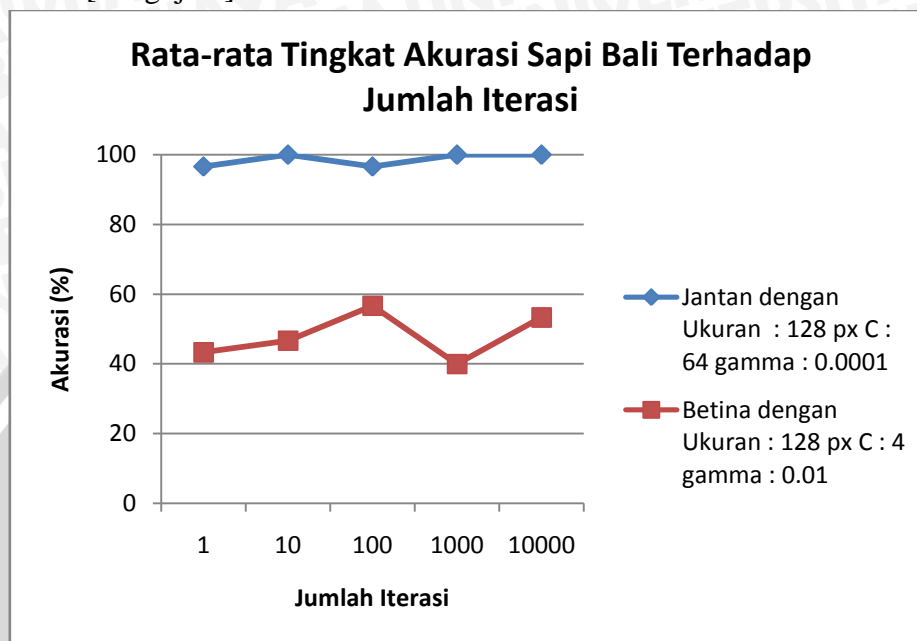
- Akurasi Sapi Bali betina dengan ukuran *cropping* 128 px, rasio 90:10, C = 4 dan *gamma* ( $\gamma$ ) = 0.01

Tabel 6.13 Tingkat Akurasi Sapi Bali Betina Terhadap Pengaruh Jumlah Iterasi (%)

Iterasi	Hasil Percobaan ke- (%)					Rata-rata(%)
	1	2	3	4	5	
1	33.33	66.67	33.33	33.33	50	43.33

10	50	50	50	50	33.33	46.67
100	66.67	66.67	66.67	33.33	50	56.67
1000	50	33.33	33.33	66.67	16.67	40
10000	33.33	16.67	83.33	66.67	66.67	53.33

Sumber : [Penguujian]



Gambar 6.6 Grafik Tingkat Akurasi Sapi Bali Terhadap Pengaruh Jumlah Iterasi ( $\gamma$ )  
 Sumber : [Penguujian]

Berdasarkan hasil pengujian terhadap jumlah iterasi didapatkan rata-rata tingkat akurasi tertinggi sebesar 100 % pada Sapi Bali jantan dengan jumlah iterasi = 10, 1000 dan 10000. Pada pengujian Sapi Bali betina tingkat akurasi tertinggi terdapat pada jumlah iterasi = 100 sebesar 56.67%. Dengan demikian penggunaan jumlah iterasi untuk pengujian selanjutnya pada Sapi Bali jantan menggunakan jumlah iterasi 10 dikarenakan proses perhitungan komputasi pada jumlah iterasi 10 lebih cepat dibandingkan jumlah iterasi 1000 dan 10000 sedangkan pada Sapi Bali betina menggunakan jumlah iterasi 100.

#### 6.1.4.3 Analisis Skenario Pengujian 4

Pada pengujian keempat Sapi Bali jantan dan Sapi Bali betina untuk tingkat akurasi terhadap pengaruh jumlah iterasi memiliki tingkat akurasi tertinggi sebesar 100% dan 76.67%. Berdasarkan tabel 6.12 pada Sapi Bali jantan dapat dilihat bahwa ketika jumlah iterasi dinaikkan ataupun diturunkan dari jumlah iterasi 100 tidak terjadi penurunan rata-rata tingkat akurasi. Pada jumlah iterasi 1000 dan 10000 menghasilkan rata-rata tingkat akurasi sebesar 100% sedangkan pada jumlah iterasi 1

menghasilkan rata-rata tingkat akurasi yang sama dengan jumlah iterasi 100 sebesar 96.67%. Berdasarkan tabel 6.13 untuk Sapi Bali betina dapat dilihat bahwa ketika jumlah iterasi dinaikkan ataupun diturunkan dari jumlah iterasi 100 tidak terjadi peningkatan rata-rata tingkat akurasi. Pada jumlah iterasi 1000 dan 10000 menghasilkan rata-rata tingkat akurasi sebesar 40% dan 53.33% sedangkan pada jumlah iterasi 10 dan 1 menghasilkan rata-rata tingkat akurasi sebesar 46.67% dan 43.33%.

Hal ini dapat disimpulkan bahwa penggunaan jumlah iterasi pada algoritma *Simplified SMO* tidak memiliki pengaruh yang signifikan terhadap proses klasifikasi dikarenakan pada algoritma *Simplified SMO* terdapat sifat stokastik yang dimana pada saat penentuan data latih pada proses pelatihan dilakukan secara acak (*random*) sehingga tingkat akurasi yang dihasilkan sangat dipengaruhi oleh penentuan data latih yang tepat.

## **6.2 Pengujian Akurasi**

Pada subbab ini akan menjelaskan tentang tingkat akurasi yang dihasilkan dari 15 data uji untuk Sapi Bali jantan dan betina. Pengujian ini akan menggunakan kombinasi parameter terbaik yang didapatkan dari pengujian skenario sebelumnya

### **6.2.1 Pengujian Tingkat Akurasi Terhadap Nilai Klasifikasi Sistem Dan Pakar**

Pada subbab ini akan menjelaskan tentang pengujian tingkat akurasi terhadap data uji, hasil skenario serta analisis hasil dari skenario pengujian tingkat akurasi terhadap nilai klasifikasi sistem dan pakar.

#### **6.2.1.1 Skenario Pengujian**

Pengujian ini menggunakan parameter terbaik yang didapatkan pada skenario pengujian sebelumnya yaitu ukuran *cropping* 128 px, rasio 90:10,  $C = 64$ , *gamma* ( $\gamma$ ) = 0.0001 dan jumlah iterasi = 10 untuk Sapi Bali jantan sedangkan pada Sapi Bali betina yaitu ukuran *cropping* 128 px, rasio 90:10,  $C = 4$ , *gamma* ( $\gamma$ ) = 0.01 dan jumlah iterasi = 1000. Data uji yang digunakan merupakan data uji yang diambil dari dataset Sapi Bali tetapi pada proses pengambilan *sample* warna kulit pada data uji berbeda dari dataset.

#### **6.2.1.2 Hasil Skenario Pengujian**

Pengujian ini akan menggunakan 15 data uji pada masing-masing jenis kelamin dan tiap kelas. Berikut ini adalah hasil pengujian data uji pada Sapi Bali jantan dan betina.

### A. Sapi Bali Jantan

- Tingkat akurasi pengujian pada Sapi Bali jantan dengan ukuran *cropping* 128 px, rasio 90:10,  $C = 64$ ,  $\text{gamma} (\gamma) = 0.0001$  dan jumlah iterasi = 10.

Tabel 6.14 Tingkat Akurasi Sapi Bali Jantan

No	Id Sapi	Fitur			Kelas	Prediksi	Status
		Red	Green	Blue			
1	SB-13	18	22.33	23.67	Baik	Baik	B
2	SB-15	17.33	20.67	20.67	Baik	Baik	B
3	SB-14	91	92.67	102	Baik	Baik	B
4	SB-6	90	80.67	93.33	Baik	Baik	B
5	SB-2	137.33	128.33	128.67	Baik	Sedang	S
6	SB-19	117.33	96	80.67	Sedang	Sedang	B
7	SB-26	101.33	78.33	60	Sedang	Sedang	B
8	SB-28	109.33	80	58.67	Sedang	Sedang	B
9	SB-25	123	98	76	Sedang	Sedang	B
10	SB-29	118.33	100.33	83.33	Sedang	Sedang	B
11	SB-34	209.33	152.66	98.33	Buruk	Buruk	B
12	SB-37	158.66	117	81.67	Buruk	Buruk	B
13	SB-35	184	134	86.67	Buruk	Buruk	B
14	SB-39	166	125.67	90.67	Buruk	Buruk	B
15	SB-32	208.33	150.33	101.67	Buruk	Buruk	B

Sumber : [Pengujian][Lampiran 6]

$$\begin{aligned} \text{Rata-rata} &= \frac{\text{Jumlah data Sapi Bali yang benar}}{\text{Total data Sapi Bali}} \times 100\% \\ &= \frac{14}{15} \times 100\% = \mathbf{93.33\%} \end{aligned}$$

### B. Sapi Bali Betina

- Tingkat akurasi pengujian pada Sapi Bali betina dengan ukuran *cropping* 128 px, rasio 90:10,  $C = 4$ ,  $\text{gamma} (\gamma) = 0.01$  dan jumlah iterasi = 1000.

Tabel 6.15 Tingkat Akurasi Sapi Bali Betina

No	Id Sapi	Fitur			Kelas	Prediksi	Status
		Red	Green	Blue			
1	SB-49	86.67	56.67	34.33	Baik	Baik	B
2	SB-56	198.33	138.33	88	Baik	Baik	B
3	SB-50	113.67	87.67	63.67	Baik	Baik	B
4	SB-60	103	62	39	Baik	Baik	B
5	SB-51	151.67	113.33	95.33	Baik	Baik	B
6	SB-64	192.33	134	78	Sedang	Sedang	B
7	SB-73	218.33	152	101.33	Sedang	Sedang	B
8	SB-66	202	142.33	85.67	Sedang	Baik	S

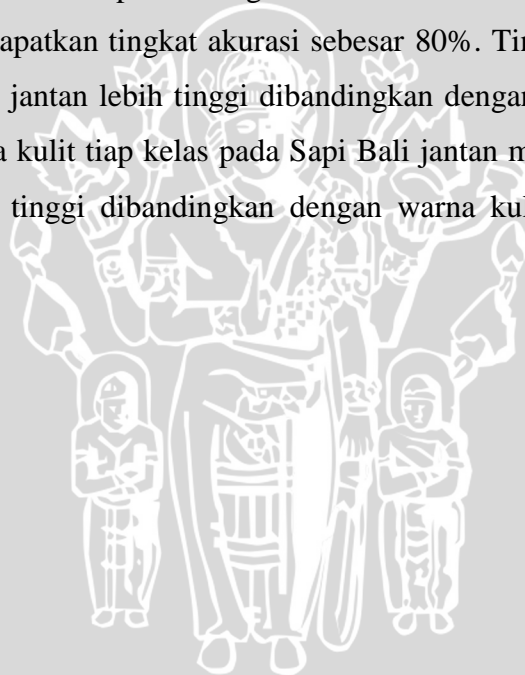
9	SB-74	197.33	141	98.66	Sedang	Baik	S
10	SB-69	206.67	141.67	77	Sedang	Sedang	B
11	SB-78	113.33	90.33	73	Buruk	Buruk	B
12	SB-83	204.33	162	118.67	Buruk	Buruk	B
13	SB-89	154.67	120.67	78.33	Buruk	Buruk	B
14	SB-77	113.67	74.33	91.67	Buruk	Buruk	B
15	SB-90	148.67	118	106.33	Buruk	Baik	S

Sumber : [Pengujian][Lampiran 6]

$$\begin{aligned} \text{Rata-rata} &= \frac{\text{Jumlah data Sapi Bali yang benar}}{\text{Total data Sapi Bali}} \times 100\% \\ &= \frac{12}{15} \times 100\% = \mathbf{80\%} \end{aligned}$$

### 6.2.1.3 Analisis Skenario Pengujian

Pada pengujian kelima menggunakan 15 data uji untuk Sapi Bali jantan dan betina. Pada Sapi Bali jantan didapatkan tingkat akurasi sebesar 93.33% sedangkan pada Sapi Bali betina didapatkan tingkat akurasi sebesar 80%. Tingkat akurasi yang dihasilkan oleh Sapi Bali jantan lebih tinggi dibandingkan dengan Sapi Bali betina, hal ini dikarenakan warna kulit tiap kelas pada Sapi Bali jantan memiliki perbedaan warna kulit yang sangat tinggi dibandingkan dengan warna kulit pada Sapi Bali betina.



## BAB VII PENUTUP

### 7.1 Kesimpulan

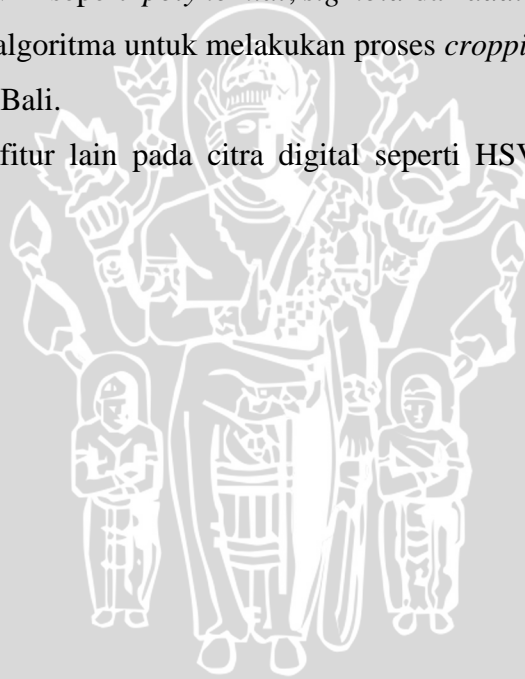
Dari hasil analisis dan pengujian yang telah dilakukan maka kesimpulan yang dapat diambil dari penelitian ini adalah:

1. Pemilihan bibit unggul Sapi Bali dapat diimplementasikan menggunakan algoritma *Simplified Sequential Minimal Optimization* (SSMO) pada Support Vector Machine (SVM) dengan kernel *Radial Basis Function* (RBF), dimana sistem terdiri dari tiga bagian utama yaitu memasukkan data, proses pelatihan dan proses pengujian.
2. Tingkat akurasi tertinggi Sapi Bali jantan betina pada pengujian pertama sebesar 80% dan 60% dengan rasio data latih dan data uji 90:10 dan ukuran *cropping* 128×128 px. Pengujian kedua tingkat akurasi tertinggi pada Sapi Bali jantan sebesar 86.67% dengan nilai *Complexity* (C) 64 dan Sapi Bali betina sebesar 60% dengan nilai *Complexity* (C) 4. Pengujian ketiga tingkat akurasi tertinggi pada Sapi Bali Jantan sebesar 96.67% dengan nilai *gamma* ( $\gamma$ ) 0.0001 dan Sapi Bali betina sebesar 46.67% dengan nilai *gamma* ( $\gamma$ ) 0.01. Pada pengujian keempat tingkat akurasi tertinggi pada Sapi Bali jantan sebesar 100% pada jumlah iterasi 10, 1000, 10000 dan Sapi Bali betina sebesar 56.67% pada jumlah iterasi 100. Kombinasi parameter terbaik pada penggunaan algoritma *Simplified SMO* pada Sapi Bali jantan yaitu rasio 90:10, ukuran *cropping* 128×128 px, *Complexity* (C) 64, *gamma* ( $\gamma$ ) 0.0001 dan jumlah iterasi 10. Pada Sapi Bali betina yaitu rasio 90:10, ukuran *cropping* 128×128 px, *Complexity* (C) 4, *gamma* ( $\gamma$ ) 0.01 dan jumlah iterasi 1000.
3. Hasil pengujian terhadap 15 data uji didapatkan tingkat akurasi sebesar 93.33% pada Sapi Bali jantan sedangkan pada Sapi Bali betina sebesar 80%. Tingkat akurasi yang dihasilkan sangat dipengaruhi oleh citra Sapi Bali yang didapatkan pada saat observasi dikarenakan terdapat *noise* seperti bayangan (*shadow*), hewan (lalat), kotoran dan cahaya yang terlalu terang pada citra Sapi Bali.

## 7.2 Saran

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

1. Diharapkan untuk pengambilan data citra Sapi Bali di lapangan menggunakan alat bantu kamera beurpa tele yang berkualitas bagus sehingga dapat memudahkan dan memberikan hasil citra 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 lain yang dapat diterapkan pada metode SVM seperti *polynomial*, *sigmoid* dan *additive kernel*.
4. Menggunakan algoritma untuk melakukan proses *cropping* secara otomatis pada citra Sapi Bali.
5. Menggunakan fitur lain pada citra digital seperti HSV, HSI, YUV dan YCbCr.





## DAFTAR PUSTAKA

- [1] 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.
- [2] Aghbari, Zaher., Makinouchi, Akifumi. 2003. "*Semantic Apporach to Image Database Classification and Retrieval*", NII Journal, No.7, Kyushu University
- [3] Buono, Agus., Herdiyeni, Yeni., Noorniawati, Vita Y. 2007. "*Klasifikasi Citra Dengan Support Vector Machine Pada Sistem Tenu Kembali Citra*". Seminar Nasional Sistem dan Informatika. Bali
- [4] Bandini, Y. 2003. Sapi Bali. Penebar Swadaya. Jakarta
- [5] Campbell, Colin., Ying, Yiming. 2011. "*Learning with Support Vector Machines*". Morgan & Claypool. United States.
- [6] 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.
- [7] Gonzalez, R.C. dan R.E. Woods, 2002. *Digital Image Processing, 2<sup>nd</sup> ed.*, Prentice-Hall Inc., New Jersey.
- [8] Guntoro, S. 2002. *Membudidayakan Sapi Bali*. Kanisius, Yogyakarta.
- [9] Guruvasuki, R. dan Arasi, Josephine Puspha. 2013. "*MRI Brain Image Retrieval Using Multi Support Vector Machine Classifier*", International Journal of Advanced Information Science and Technology (IJAIST), Vol.10, No.10, India.
- [10] Hardianto, Agung., Liliana, Dewi Y., Ridok, M. 2011. "*Indonesian News Classification using Support Vector Machine*". World Academy of Science, Engineering and Technology, Vol:5., No.9.
- [11] Hsu, Chih-Wei., Chih-Jen Lin. 2002. "*A Comparison Methods for Multi-Class Support Vector Machine*". IEEE Transactions on Neural Networks, Vol. 13, No.2, 415-425.
- [12] Hsu, Chih-Wei., Chang, Chih-Chung., Chih-Jen Lin. 2002. "*A Practical Guide to Support Vector Classification*". Departement of Computer Science, National Taiwan Univesity, Taiwan.

- [13] Han, Jiawei and Kamber, Micheline. 2006. *"Data Mining Concept and Techniques Second Edition"*. Morgan Kaufmann Publisher. San Francisco.
- [14] Kusnawi. 2007. *Pengantar Solusi Data Mining*. STMIK AMIKOM, Yogyakarta.
- [15] Munawaroh, S., Sutanto, F A. 2010. "Pengolahan Citra Digital untuk Identifikasi Uang Kertas", *Jurnal Teknologi Informasi DINAMIK*, Volume XV, No.1, hal 34-40.
- [16] Ng, Andrew. 2009. *Cs229 The Simplified SMO Algorithm*. <http://cs229.stanford.edu/materials/smo.pdf> [5 Juni 2014].
- [17] Platt, JC. 1998. *"Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machine"*. Microsoft Research.
- [18] 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.
- [19] Sembiring, Krisantus. 2007. *Penerapan Teknik Support Vector Machine untuk Pendeteksian Intrusi pada Jaringan*. Institut Teknologi Bandung, Bandung.
- [20] Turban, Efraim. Aranson, Jay, E. dan Lian, Teng-Peng. Alih bahasa oleh Prabantini, Dwi. 2005. *"Decision Support System and Intelligent System, 7<sup>th</sup> edition"*, Jilid 1, Penerbit Anda, Yogyakarta.
- [21] 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 (IJESET)*, Volume 2, Issue 2, pp: 19-26, India.
- [22] Wang, Lipo (Ed). 2005. *Support Vector Machines: Theory and Application*. Springer, New York.