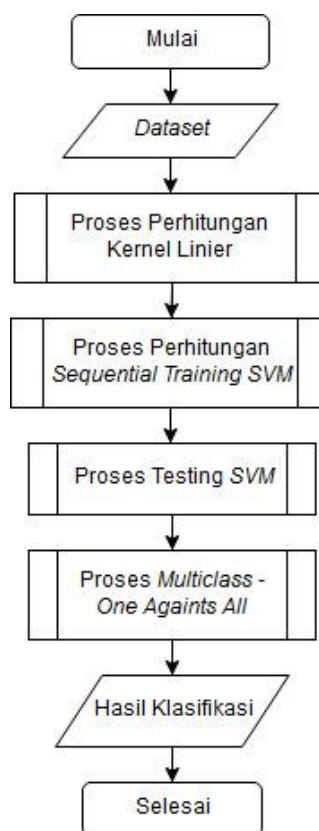


BAB 4 ANALISIS DAN PERANCANGAN

Analisis dan Perancangan menjelaskan tentang bagaimana cara untuk melakukan klasifikasi kualitas susu sapi dengan menggunakan metode *Support Vector Machine* sehingga dapat dibangun ke dalam sistem *prototype* yang akan dihasilkan.

Pada sub bab ini menjelaskan mengenai penyelesaian masalah yang terdapat pada klasifikasi kualitas susu sapi dengan menggunakan metode *Support Vector Machine*. Pada perancangan alur penyelesaian masalah ini akan dijelaskan detail setiap proses yang diperlukan.

4.1 Proses *Support Vector Machine (SVM)*



Gambar 4.1 Diagram Alir Proses *Support Vector Machine*

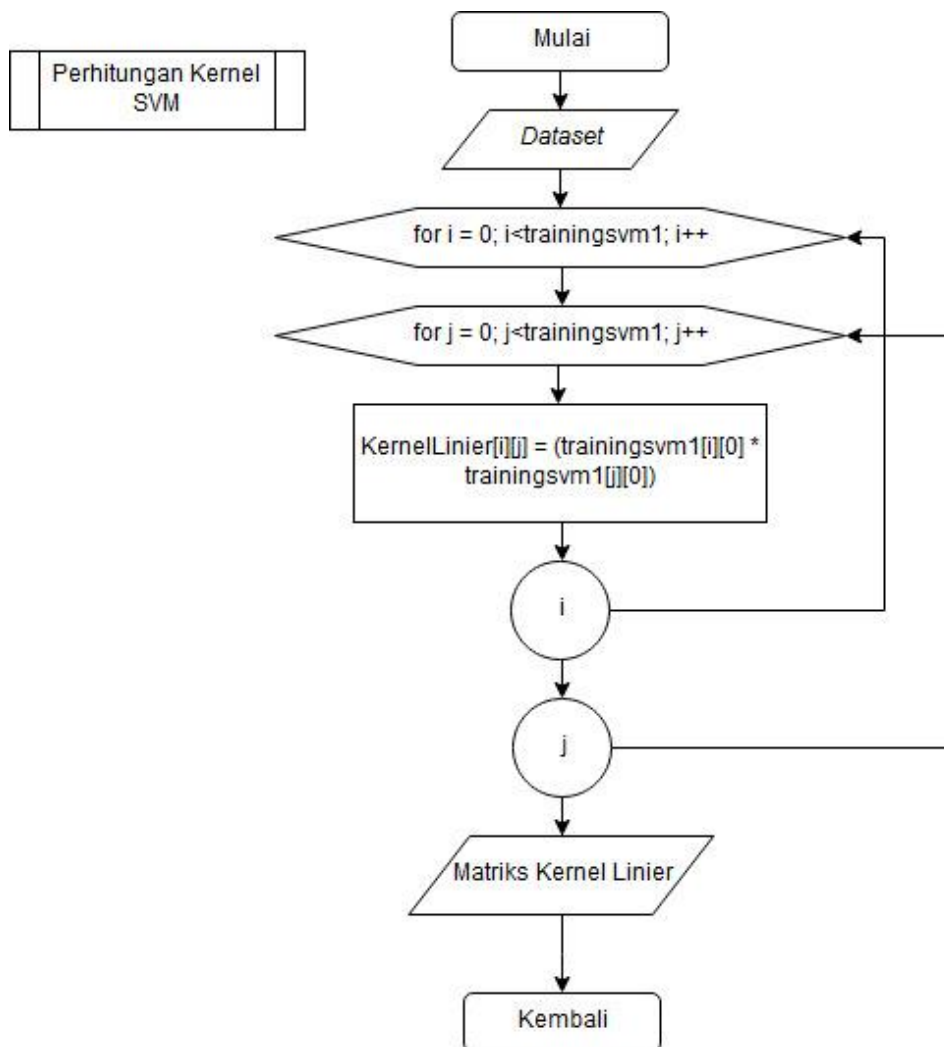
Pada Gambar 4.1 menjelaskan tahapan proses *Support Vector Machine*.

1. Melakukan *input dataset*.
2. Melakukan perhitungan *kernel Linier SVM*.
3. Melakukan perhitungan data *training SVM*, menggunakan metode *sequential training SVM* diantaranya seperti berikut:

- a. Menghitung Matrix Hessian
 - b. Menghitung nilai E_i
 - c. Menghitung nilai $i \delta \alpha_i$
 - d. Menentukan nilai α_i
4. Melakukan perhitungan data *testing* untuk dilakukan *testing*.
 5. Melakukan perhitungan *multiclass One Against All*
 6. *Output* yang dihasilkan adalah *predicted class*.

4.1.1 Proses Kernel Linier

Langkah proses kernel linier dapat dilihat pada Gambar 4.2 Diagram Alir Perhitungan Kernel SVM Linier.



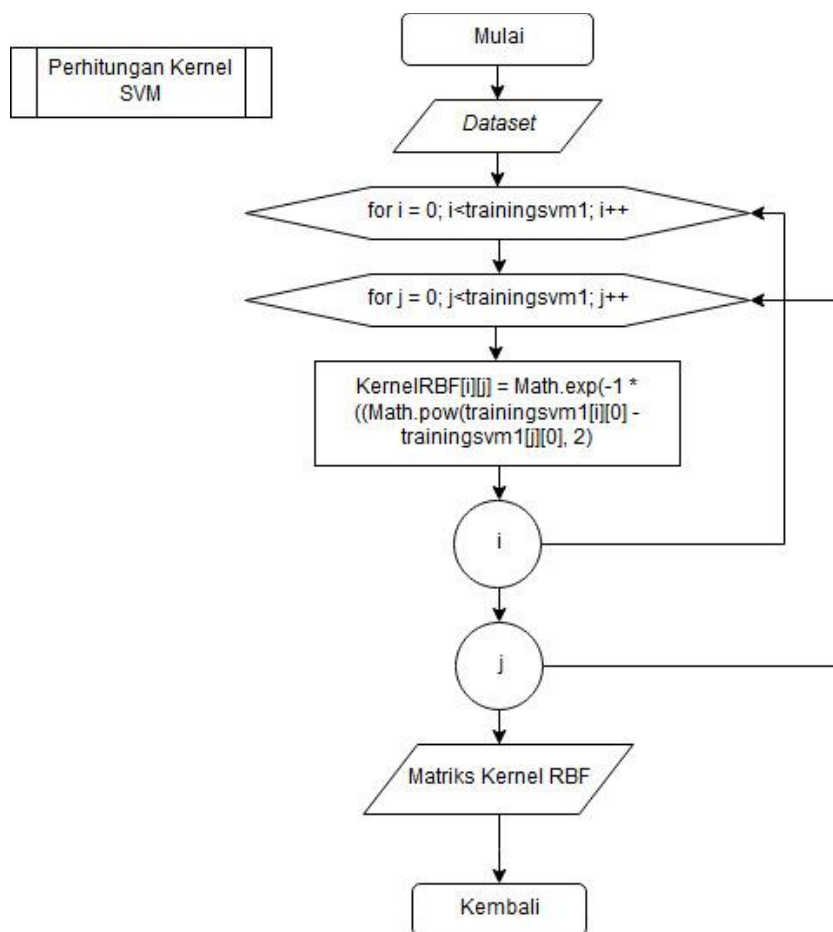
Gambar 4.2 Diagram Alir Perhitungan Kernel SVM Linier

Pada Gambar 4.2 menjelaskan tentang tahapan proses perhitungan kernel SVM Linier. Berikut adalah penjelasannya.

1. Melakukan *input dataset*.
2. Melakukan perulangan terhadap banyaknya jumlah dataset.
3. Melakukan proses perhitungan Kernel Linier.
4. Output yang dihasilkan adalah matriks kernel Linier.

4.1.2 Proses Kernel Radial Basis Function (RBF)

Langkah proses kernel linier dapat dilihat pada Gambar 4.3 Diagram Alir Perhitungan Kernel SVM RBF.



Gambar 4.3 Diagram Alir Perhitungan Kernel SVM RBF

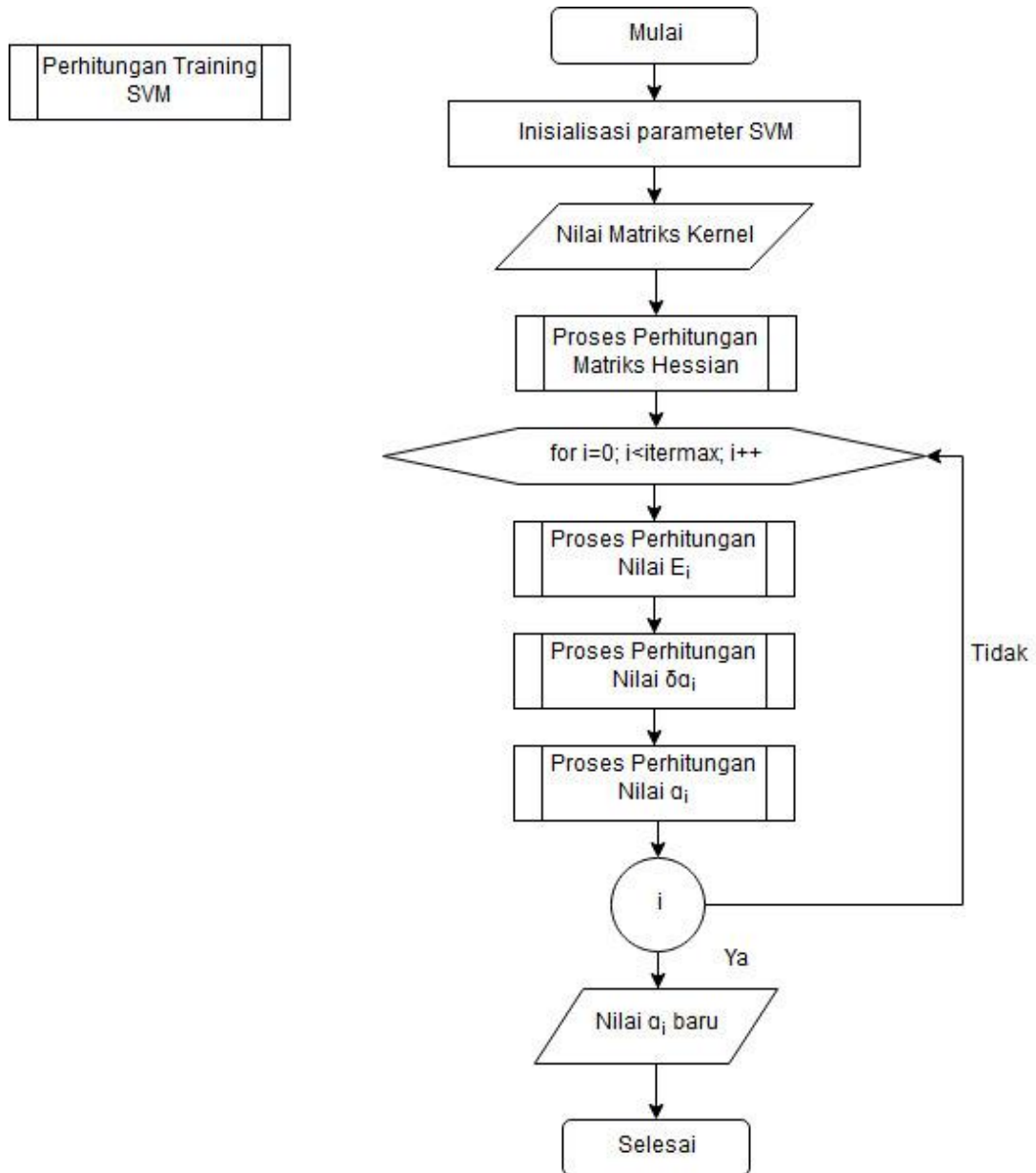
Pada Gambar 4.3 menjelaskan tentang tahapan proses perhitungan kernel SVM Linier. Berikut adalah Penjelasannya.

1. Melakukan *input dataset*.

2. Melakukan perulangan terhadap banyaknya jumlah dataset.
3. Melakukan proses perhitungan Kernel Linier.
4. Output yang dihasilkan adalah matriks kernel Linier.

4.1.3 Perhitungan Sequential Training SVM

Langkah perhitungan *sequential training* dapat dilihat pada Gambar 4.4 Diagram Alir Perhitungan *Sequential Training* SVM.



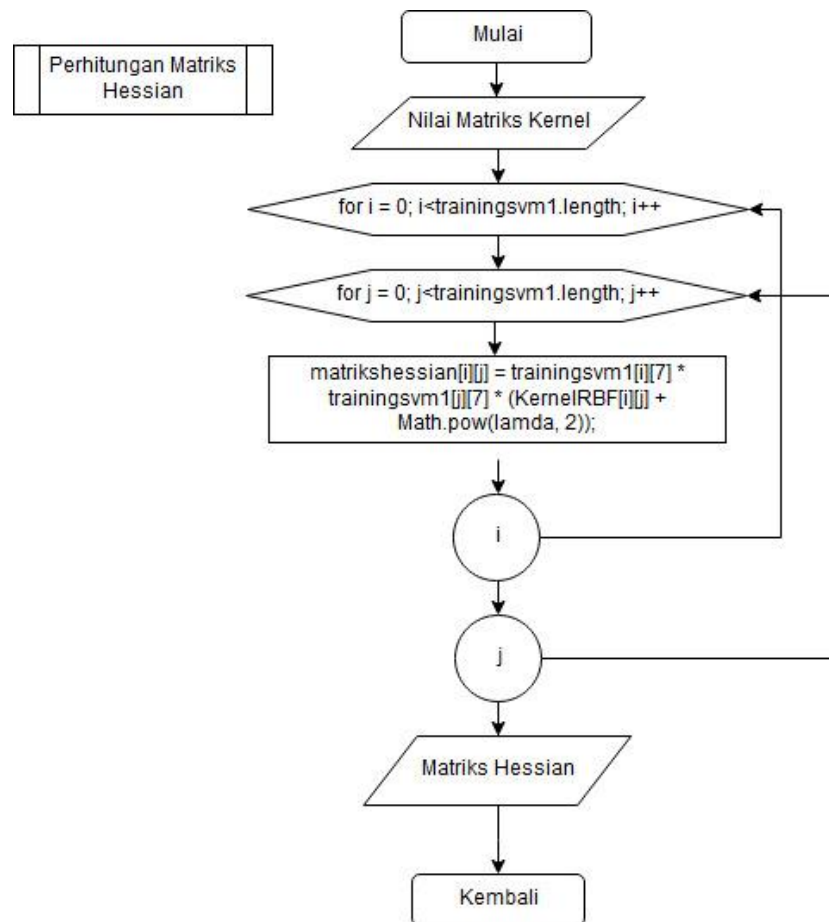
Gambar 4.4 Diagram Alir Perhitungan *Sequential Training* SVM

Pada Gambar 4.4 dijelaskan tahapan-tahapan perhitungan *Sequential Training* SVM. Berikut adalah penjelasannya.

1. Meng-*input*-kan nilai matriks kernel.
2. Melakukan inisialisasi parameter SVM.
3. Melakukan perhitungan *Matriks Hessian*.
4. Melakukan perhitungan nilai E_i .
5. Melakukan perhitungan nilai $\delta\alpha_i$.
6. Melakukan perhitungan nilai α_i .
7. Melakukan proses iterasi sampai nilai konvergen ($|\delta\alpha_i| < \epsilon$) atau telah mencapai nilai itermax kemudian berhenti.
8. *Output* yang dihasilkan adalah nilai α_i .

4.1.4 Perhitungan Matriks Hessian

Langkah perhitungan nilai matriks *hessian* dapat dilihat pada Gambar 4.5 Diagram Alir Perhitungan Matriks Hessian.



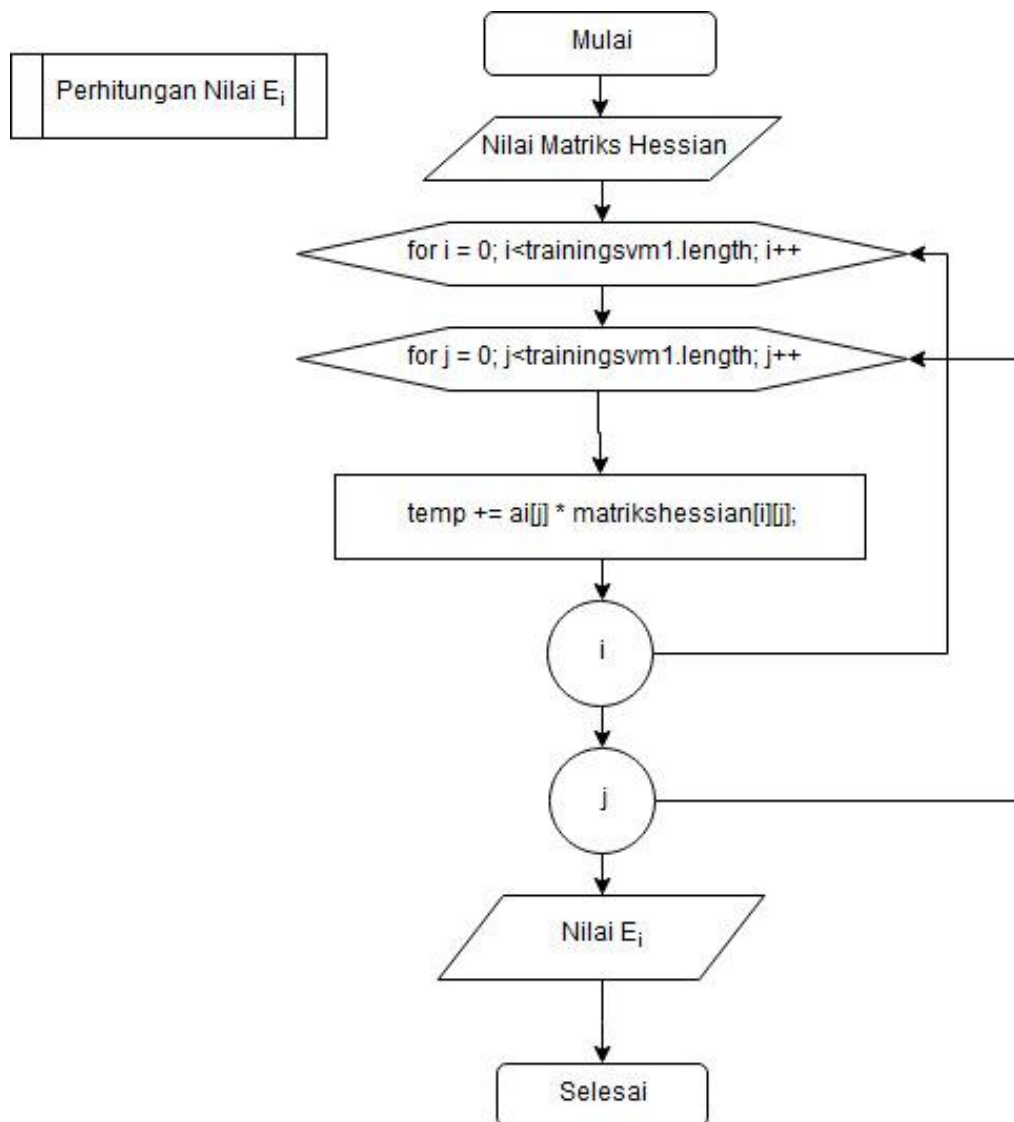
Gambar 4.5 Diagram Alir Perhitungan nilai Matriks Hessian

Pada Gambar 4.5 dijelaskan tahapan-tahapan perhitungan nilai matriks *hessian*. Berikut adalah penjelasannya.

1. Meng-*input*-kan nilai matriks kernel.
2. Melakukan perhitungan matriks *hessian*.
3. *Output* yang dihasilkan adalah nilai matriks *hessian*.

4.1.5 Perhitungan Nilai E_i

Langkah perhitungan nilai E_i dapat dilihat pada Gambar 4.6 Diagram Alir Perhitungan Nilai E_i .



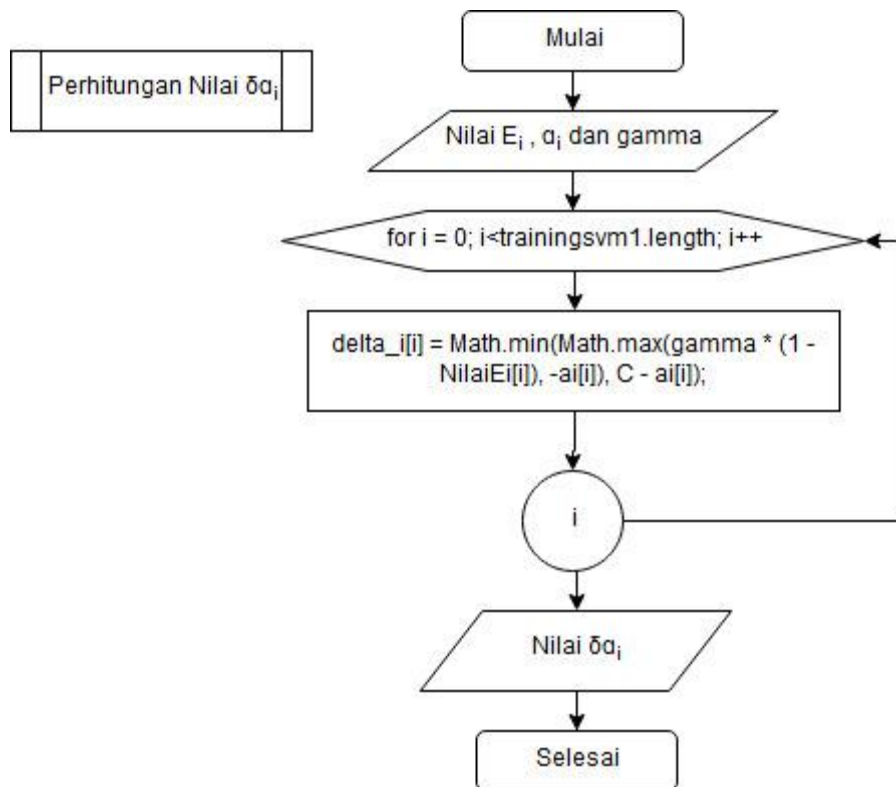
Gambar 4.6 Diagram Alir Perhitungan nilai E_i

Pada Gambar 4.6 dijelaskan tahapan-tahapan perhitungan Nilai E_i . Berikut adalah penjelasannya.

1. Meng-*input*-kan nilai matriks *hessian* yang telah didapatkan sebelumnya.
2. Inisialisasi terhadap nilai α awal 0.
3. Melakukan perhitungan nilai E_i .
4. Melakukan perulangan jumlah *dataset* dan iterasi untuk proses perhitungan nilai E_i .
5. *Output* yang dihasilkan adalah nilai E_i .

4.1.6 Perhitungan nilai $\delta\alpha_i$

Langkah perhitungan nilai $\delta\alpha_i$ dapat dilihat pada Gambar 4.7 Diagram Alir Perhitungan Nilai $\delta\alpha_i$.



Gambar 4.7 Diagram Alir Perhitungan Nilai $\delta\alpha_i$.

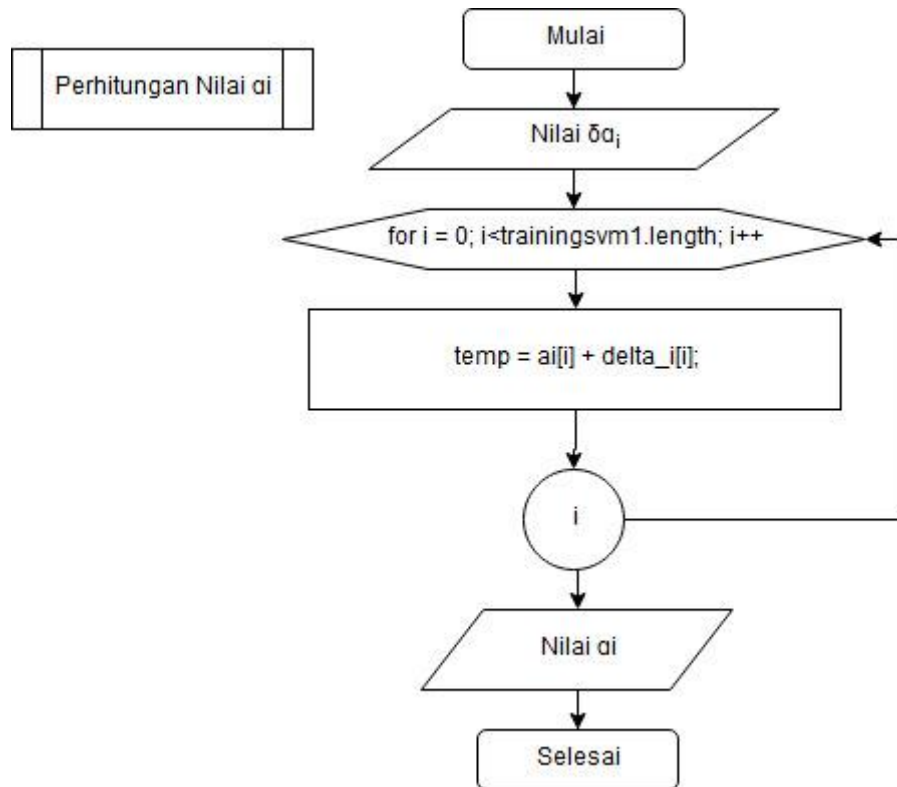
Pada Gambar 4.7 dijelaskan tahapan-tahapan perhitungan nilai $\delta\alpha_i$. Berikut adalah penjelasannya.

1. Meng-*input*-kan nilai E_i yang sudah didapatkan sebelumnya dan inisialisasi C , nilai α_i dan nilai γ .
2. Melakukan perulangan terhadap banyaknya jumlah *dataset* ke- i .

3. Melakukan perhitungan nilai δa_i .
4. *Output* yang dihasilkan adalah nilai δa_i .

4.1.7 Perhitungan Nilai a_i .

Langkah perhitungan nilai a_i dapat dilihat pada Gambar 4.8 Diagram Alir Perhitungan Nilai a_i .



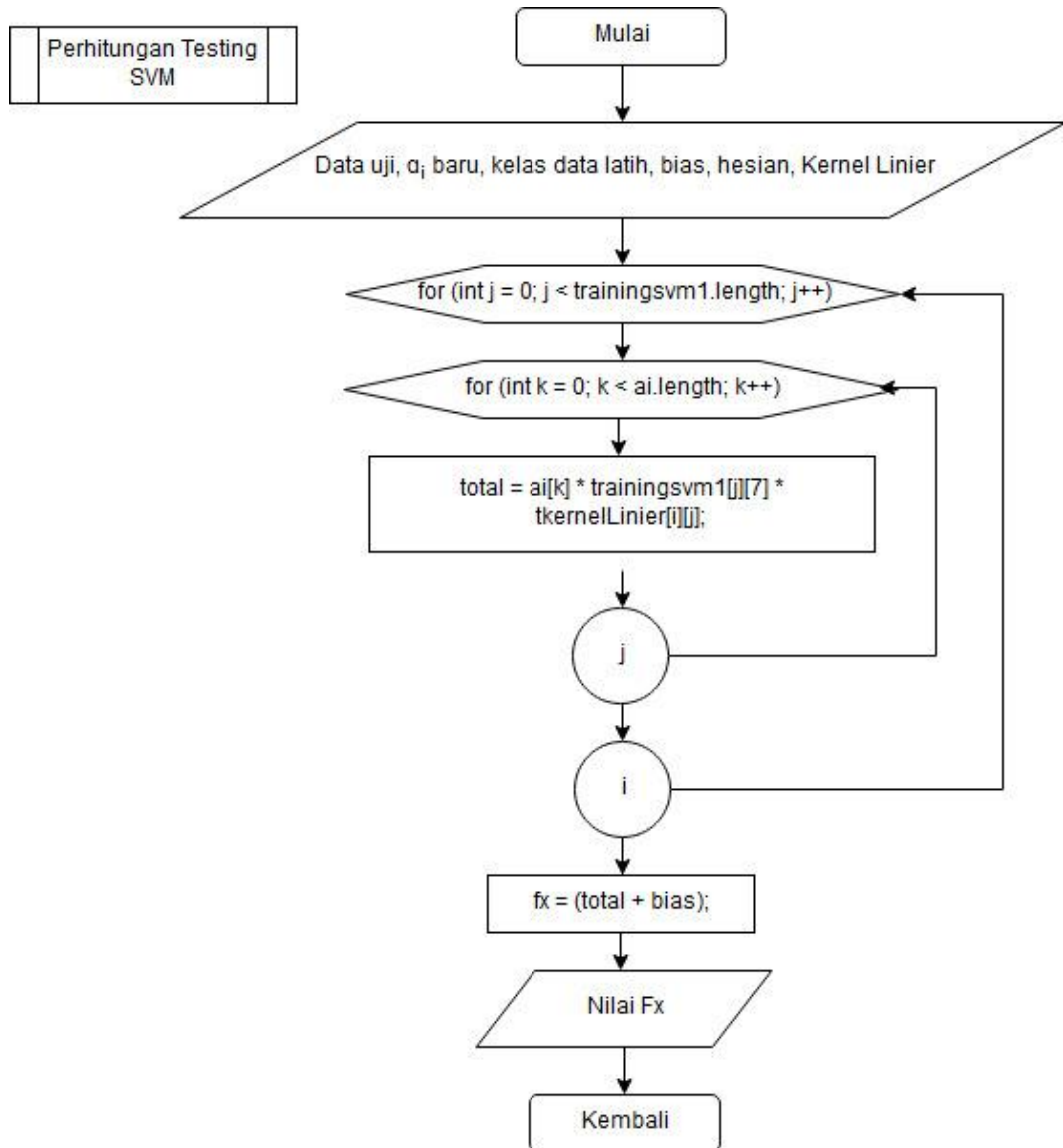
Gambar 4.8 Diagram Alir Perhitungan Nilai a_i .

Pada Gambar 4.8 dijelaskan tahapan-tahapan perhitungan nilai δa_i . Berikut adalah penjelasannya.

1. Meng-*input*-kan nilai δa_i yang sudah didapatkan sebelumnya.
2. Melakukan perulangan terhadap banyaknya jumlah *dataset* ke-*i*.
3. Melakukan perhitungan nilai a_i .
4. *Output* yang dihasilkan adalah nilai a_i .

4.1.8 Perhitungan Testing SVM Linier

Langkah perhitungan *testing* SVM Linier dapat dilihat pada Gambar 4.9 Diagram Alir Perhitungan Nilai *Testing* SVM Linier.



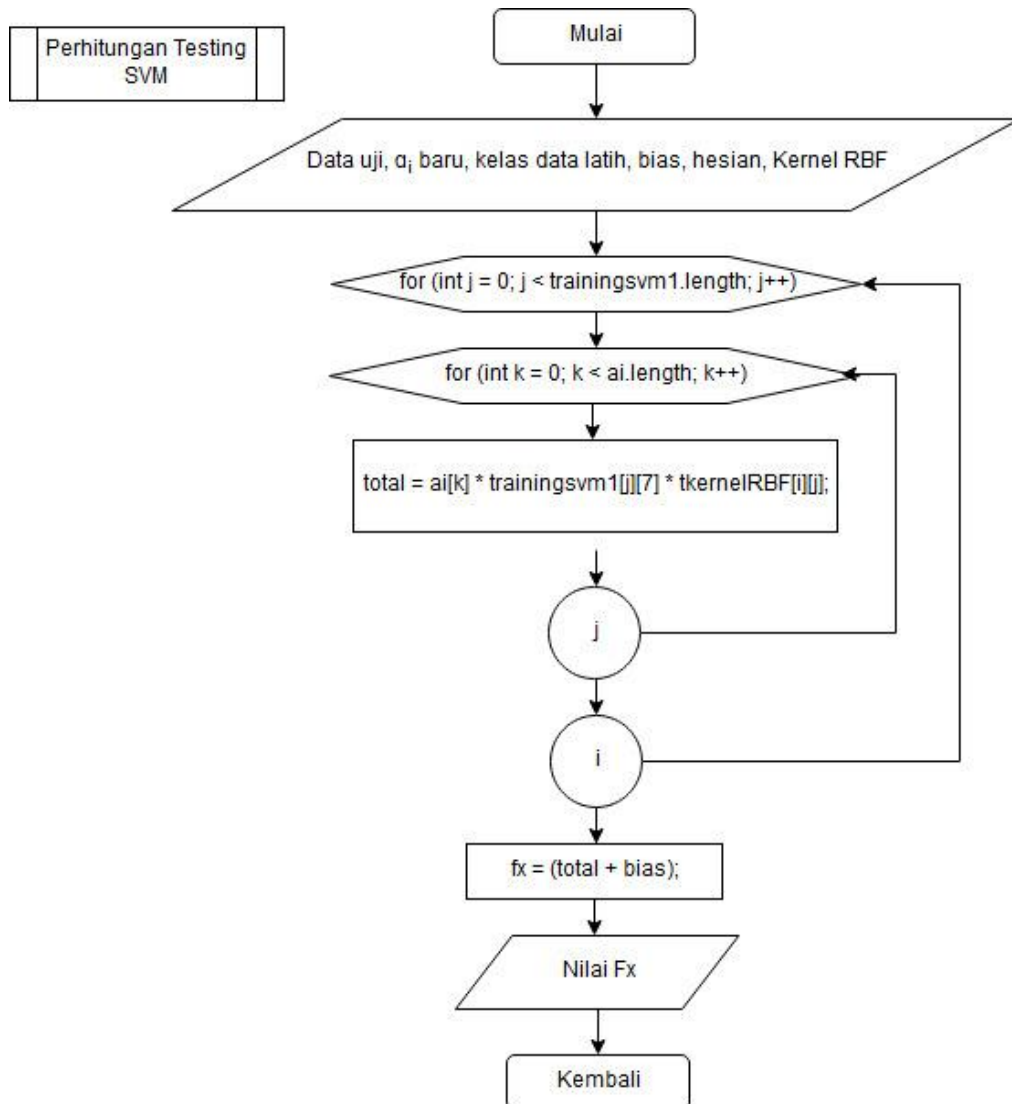
Gambar 4.9 Diagram Alir *Testing* SVM Linier

Pada Gambar 4.9 dijelaskan tahapan-tahapan perhitungan nilai testing SVM linier. Berikut adalah penjelasannya.

1. Meng-*input*-kan *dataset*, a_i baru, kelas *dataset*, bias, matriks *hesian*, kernel linier.
2. Melakukan perulangan sebanyak jumlah *dataset*.
3. Menghasilkan *Output* nilai kernel linier.
4. Melakukan perhitungan nilai fx .
5. *Output* yang dihasilkan adalah nilai fx .

4.1.9 Perhitungan Testing Svm RBF

Langkah perhitungan *testing* SVM RBF dapat dilihat pada Gambar 4.10 Diagram Alir Perhitungan Nilai *Testing* SVM RBF.



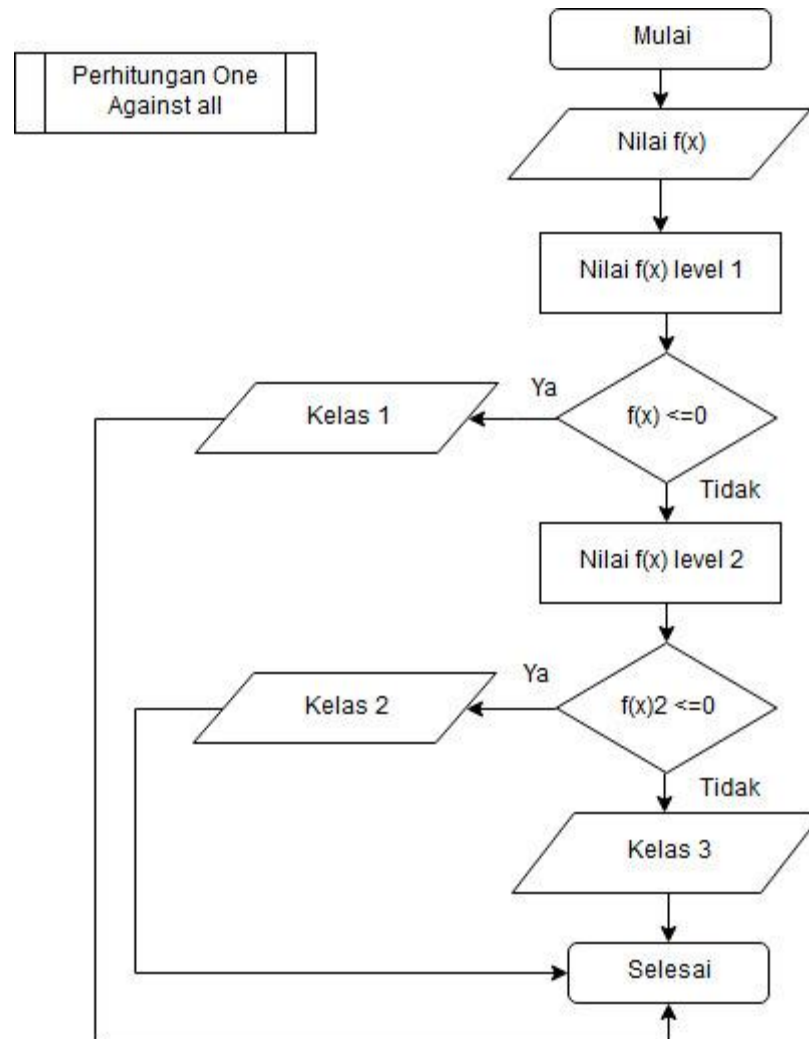
Gambar 4.10 Diagram Alir *Testing* SVM RBF

Pada Gambar 4.10 dijelaskan tahapan-tahapan perhitungan nilai testing SVM linier. Berikut adalah penjelasannya.

1. Meng-*input*-kan *dataset*, a_i baru, kelas *dataset*, bias, matriks *hesian*, kernel RBF.
2. Melakukan perulangan sebanyak jumlah *dataset*.
3. Menghasilkan *Output* nilai kernel RBF.
4. Melakukan perhitungan nilai fx .
5. *Output* yang dihasilkan adalah nilai fx .

4.1.10 Perhitungan *Multiclass One Againsts All*

Langkah perhitungan *multiclass one againsts all* dapat dilihat pada Gambar 4.11 Diagram Alir Perhitungan Nilai *Multiclass One Againsts All*.



Gambar 4.11 Diagram Alir Perhitungan Nilai *Multiclass One Againsts All*

Pada Gambar 4.11 dijelaskan tahapan-tahapan perhitungan nilai testing SVM linier. Berikut adalah penjelasannya.

1. Meng-*input*-kan nilai fx level 1.
2. Jika nilai fx level 1 bernilai ≤ 0 maka *output* yang dihasilkan adalah kelas 1 dan jika tidak maka akan dilanjutkan nilai fx level 2.
3. Jika nilai fx level 2 bernilai ≤ 0 maka *output* yang dihasilkan adalah kelas 2 dan jika tidak maka *output* yang dihasilkan adalah kelas 3.

4.2 Perhitungan Manual

Perhitungan manual pada penelitian ini menggunakan dataset susu sapi yang terdiri dari 7 atribut atau kriteria yang akan dibagi menjadi tiga kelas yaitu kualitas tinggi, standar dan rendah. Data tersebut diperoleh dari UPT Laboratorium Kesehatan Hewan Kota Malang Tahun 2015. Data yang akan digunakan untuk proses perhitungan manual sebanyak 269 data.

4.2.1 Proses Perhitungan Kernel RBF Gaussian

Pada proses perhitungan ini kernel yang digunakan adalah kernel *RBF*. Data yang digunakan untuk menghitung kernel *RBF* adalah dua pasangan data latih dan parameter σ (*sigma*). Berikut adalah proses perhitungan kernel *RBF* :

Pada Tabel 4.1 adalah tabel data *training* dengan data latih ke-1.

Data 1 (x_1)

Tabel 4.1 Data *training* 1

No.	F1	F2	F3	F4	F5	F6	F7
1	0.1	0.4382 92	0.4476 84	0.33905 3	0.32591 4	0.30660 7	0.36666 7

Misal

$k(x, y)$, misal $x = x_1, y = x_1$ maka $k(x, y) = k(x_1, x_1) =$

$$\exp\left(-\frac{\|x_1 - x_1\|^2}{2\sigma^2}\right) = \exp\left(-\frac{0}{2.1^2}\right) = \exp\left(-\frac{0}{2}\right) = \exp(0) = 1$$

$$(x_1 - x_1) = \begin{pmatrix} 0.1 \\ 0.438292 \\ 0.447684 \\ 0.339053 \\ 0.325914 \\ 0.306607 \\ 0.366667 \end{pmatrix} - \begin{pmatrix} 0.1 \\ 0.438292 \\ 0.447684 \\ 0.339053 \\ 0.325914 \\ 0.306607 \\ 0.366667 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\|x_1 - x_1\| = \sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2} = \sqrt{0} = 0$$

$$\|x_1 - x_1\|^2 = 0^2 = 0$$

Pada Tabel 4.2 adalah hasil perhitungan *Kernel Gaussian* (RBF) menggunakan Persamaan 2.15.

Tabel 4.2 Hasil perhitungan data *training* dengan *kernel* RBF

No.	A1	A2	A3	...	A13	A14	A15
A1	1	0.999997	0.99999	...	0.999988	0.999996	0.999998
A2	0.999997	1	0.999991	...	0.999981	0.999992	0.999995
A3	0.99999	0.999991	1	...	0.999975	0.999984	0.999987
...
A13	0.999988	0.999981	0.999975	...	1	0.999996	0.999995
A14	0.999996	0.999992	0.999984	...	0.999996	1	0.999999
A15	0.999998	0.999995	0.999987	...	0.999995	0.999999	1

Setelah menyelesaikan perhitungan kernel RBF, proses perhitungan selanjutnya yaitu proses perhitungan metode sequential training dimana terdapat fungsi antara lain matriks hessian, menghitung E_i , menghitung $\delta\alpha_i$, dan menghitung α_i . Sebelum memasuki perhitungan sequential training svm dibutuhkan inisialisasi parameter terlebih dahulu sebagai berikut :

1. Inisialisasi parameter $\alpha_i = 0$.
2. Inisialisasi parameter $\lambda = 0.5$.
3. Inisialisasi parameter $C = 1$.
4. Inisialisasi parameter Kons $\gamma = 0.001$.
5. Menghitung nilai $\gamma = \text{Kons } \gamma / \text{max hessian}$.

Langkah awal dari proses perhitungan sequential training svm adalah menghitung matriks hessian. (x_i, x_j) adalah *kernel* yang telah dihitung pada tahap sebelumnya. Inisialisasi nilai lamda $\lambda=0.5$. Contoh perhitungan *record* pertama dan kedua adalah sebagai berikut:

$$\begin{aligned}
 D_{11} &= y_1 y_1 (K(x_1, x_1) + \lambda^2) \\
 &= -1 \cdot -1 (1.000000 + 0.5^2) \\
 &= 1 (1.000000 + 0.25) \\
 &= 1.250000 \\
 D_{12} &= y_1 y_2 (K(x_1, x_2) + \lambda^2) \\
 &= -1 \cdot -1 (0.999996812 + 0.5^2) \\
 &= 1 (0.384139 + 0.25) \\
 &= 1.249996812
 \end{aligned}$$

Tabel 4.3 adalah hasil perhitungan *matriks hessian*.

Tabel 4.3 Hasil perhitungan *matriks hessian*

No.	A1	A2	A3	...	A13	A14	A15
A1	1.25	1.249997	1.24999	...	-0.74999	-0.75	-0.75
A2	1.249997	1.25	1.249991	...	-0.74998	-0.74999	-0.74999
A3	1.24999	1.249991	1.25	...	-0.74998	-0.74998	-0.74999
...
A13	-0.74999	-0.74998	-0.74998	...	1.25	1.249996	1.249995
A14	-0.75	-0.74999	-0.74998	...	1.249996	1.25	1.249999
A15	-0.75	-0.74999	-0.74999	...	1.249995	1.249999	1.25

Setelah menyelesaikan perhitungan matriks hessian, selanjutnya adalah menghitung E_i , δa_i , dan a_i dimana ketiga fungsi tersebut dilakukan proses iterasi sesuai dengan jumlah yang dibutuhkan.

$$\begin{aligned}
 E_i = & ((0.5 \times 1.25) + (0.5 \times 1.249997) + \\
 & (0.5 \times 1.24999) + (0.5 \times 1.249997216) + \\
 & (0.5 \times 1.249998446) + (0.5 \times 1.24999619) + \\
 & (0.5 \times 1.249996723) + (0.5 \times 1.249996723) + \\
 & (0.5 \times 1.249997551) + (0.5 \times 1.249996091) + \\
 & (0.5 \times (-0.749995871)) + (0.5 \times (-0.749990107)) + \\
 & (0.5 \times (-0.749988263)) + (0.5 \times (-0.74999622)) + \\
 & (0.5 \times (-0.749998261))) = 4.374997
 \end{aligned}$$

Tabel 4.4 Hasil perhitungan E_i

	Iterasi 1	Iterasi 2	Iterasi 3
A1	4.374997	0	4.374997
A2	4.375001	0	4.375001
A3	4.37499	0	4.37499
A4	4.374998	0	4.374998
A5	4.374997	0	4.374997
A6	4.374987	0	4.374987
A7	4.374988	0	4.374988
A8	4.374979	0	4.374979
A9	4.374989	0	4.374989
A10	4.374983	0	4.374983
A11	-0.62498	-1.24997	-1.24997
A12	-0.62497	-1.24993	-1.24993
A13	-0.62495	-1.24991	-1.24991
A14	-0.62498	-1.24996	-1.24996
A15	-0.62499	-1.24998	-1.24998

Setelah mendapatkan nilai E_i , selanjutnya adalah menghitung nilai $\delta\alpha_i$:

$$\begin{aligned} \delta\alpha_1 &= \min\{\text{maks}[\gamma(1 - E_1), -\alpha_1], C - \alpha_1\} \\ &= \min\{\text{maks}[0.5(1 - 4.374997), -0.5], 1 - 0.5\} \\ &= \min\{\text{maks}[0.5(-4.374997), -0.5], 0.5\} \\ &= \min\{\text{maks}[-4.374997, -0.5], 0.5\} \\ &= \min\{-4.374997, -0.5\} \\ &= -0.5 \end{aligned}$$

Tabel 4.5 Hasil perhitungan $\delta\alpha_i$

	Iterasi 1	Iterasi 2	Iterasi 3
A1	-0.5	0.5	-0.5
A2	-0.5	0.5	-0.5
A3	-0.5	0.5	-0.5
A4	-0.5	0.5	-0.5
A5	-0.5	0.5	-0.5
A6	-0.5	0.5	-0.5
A7	-0.5	0.5	-0.5
A8	-0.5	0.5	-0.5
A9	-0.5	0.5	-0.5
A10	-0.5	0.5	-0.5
A11	0.5	0	0
A12	0.5	0	0
A13	0.5	0	0
A14	0.5	0	0
A15	0.5	0	0

Setelah mendapatkan nilai $\delta\alpha_i$, selanjutnya adalah menghitung nilai α_i dengan fungsi sebagai berikut :

$$\begin{aligned} \alpha_1 &= \alpha_1 + \delta\alpha_1 \\ &= 0.5 + (-0.5) \\ &= 0 \end{aligned}$$

Tabel 4.6 Hasil perhitungan α_i

	Iterasi 1	Iterasi 2
A1	0	0.5
A2	0	0.5
A3	0	0.5
A4	0	0.5

A5	0	0.5
A6	0	0.5
A7	0	0.5
A8	0	0.5
A9	0	0.5
A10	0	0.5
A11	1	1
A12	1	1
A13	1	1
A14	1	1
A15	1	1

Setelah melakukan perhitungan sequential training svm, selanjutnya adalah mencari nilai $w \cdot x, x^+$ dan $w \cdot x, x^-$ dengan fungsi sebagai berikut :

$$w = \sum_{i=1}^n \alpha_i y_i x_i \quad w \cdot x^+ = \sum_{i=1}^n \alpha_i y_i x_i x^+ = \sum_{i=1}^n \alpha_i y_i k(x_i, x^+)$$

Untuk hasil perhitungan secara keseluruhan untuk mendapatkan nilai $w \cdot x^+$ dan nilai $w \cdot x^-$ dapat dilihat pada Tabel 4.9.

Tabel 4.7 Hasil perhitungan $w \cdot x^+$ dan $w \cdot x^-$

w+	w-	
0.999996	1	-1
0.999992	0.999997	-0.99999
0.999986	0.99999	-0.99999
0.999993	0.999997	-0.99999
0.999998	0.999998	-1
0.999998	0.999996	-1
0.999999	0.999997	-1
0.999999	0.999993	-1
0.999998	0.999998	-1
1	0.999996	-1
1	0.999996	1.999996
0.999997	0.99999	1.999987
0.999998	0.999988	1.999986
0.999999	0.999996	1.999995
0.999999	0.999998	1.999997

Langkah selanjutnya adalah mencari nilai b .

$$b = -\frac{1}{2}(w \cdot x^+ + w \cdot x^-)$$

$$=-7.9885$$

4.2.2 Proses Perhitungan Kernel Linier

Pada proses perhitungan kernel yang digunakan adalah kernel *Linier*. Data yang digunakan untuk menghitung kernel *Linier* adalah dua pasangan data latih beserta atributnya. Berikut adalah proses perhitungan kernel *Linier* :

1. Menginisiasi awal untuk nilai α , C, epsilon, gamma, dan lamda $\alpha=0$, C = 1, epsilon = 0.001, gamma = 0.5, lamda=0.5
2. Memasukkan data latih

Pada Tabel 4.10 adalah tabel data *training*

Tabel 4.8 Data training

No.	Fat	SNF	Density	Protein	Lactosa	Water	Temp °C
A1	0.1	0.438292	0.447684	0.339053	0.325914	0.306607	0.526667
A2	0.211111	0.16281	0.149203	0.121302	0.275415	0.26016	0.686667
...
A14	0.496296	0.525344	0.683624	0.471598	0.428239	0.1	0.633333
A15	0.37963	0.536364	0.417747	0.443195	0.482724	0.184084	0.526667

Pada metode kernel, data tidak direpresentasikan secara individual, melainkan lewat perbandingan antara sepasang data. Setiap data akan dibandingkan dengan dirinya dan data lainnya.

Pada Tabel 4.11 adalah Tabel perbandingan data.

Tabel 4.9 Tabel perbandingan data

No.	A1	A2	A3	A4	A5
A1	K(A1,A1)	K(A1,A2)	K(A1,A3)	K(A1,A4)	K(A1,A5)
A2	K(A2,A1)	K(A2,A2)	K(A2,A3)	K(A2,A4)	K(A2,A5)
A3	K(A3,A1)	K(A3,A2)	K(A3,A3)	K(A3,A4)	K(A3,A5)
A4	K(A4,A1)	K(A4,A2)	K(A4,A3)	K(A4,A4)	K(A4,A5)
A5	K(A5,A1)	K(A5,A2)	K(A5,A3)	K(A5,A4)	K(A5,A5)

Berikut adalah contoh perhitungan data A1 dengan data A1:

$$K(A1,A1) = ((0.1*0.1) + (0.438292*0.438292) + (0.447684*0.447684) + (0.339053*0.339053) + (0.325914*0.325914) + (0.306607*0.306607) + (0.526667*0.526667)) = 0.995083$$

Begitu seterusnya dilakukan perhitungan yang sama sampai K(A15,A15).

Pada Tabel 4.12 adalah hasil perhitungan *Kernel Linier*.

Tabel 4.10 Hasil perhitungan data *training* dengan *kernel* Linier

No.	A1	A2	...	A14	A15
A1	0.995083	0.731566	...	1.249613	1.101478
A2	0.731566	0.723099	...	0.928358	0.826044
...
A14	1.249613	0.928358	...	1.806542	1.523462
A15	1.101478	0.826044	...	1.523462	1.347027

Setelah menyelesaikan perhitungan kernel Linier, proses perhitungan selanjutnya yaitu proses perhitungan metode sequential training dimana terdapat fungsi antara lain matriks hessian, menghitung E_i , menghitung $\delta\alpha_i$, dan menghitung α_i . Sebelum memasuki perhitungan sequential training svm dibutuhkan inialisasi parameter terlebih dahulu sebagai berikut :

1. Inialisasi parameter $\alpha_i = 0.5$
2. Inialisasi parameter $\lambda = 0.5$
3. Inialisasi parameter $C = 1$
4. Inialisasi parameter $\text{Kons } \gamma = 0.001$.
5. Menghitung nilai $\gamma = \text{Kons } \gamma / \text{max hessian}$.

Langkah awal dari proses perhitungan sequential training svm adalah menghitung matriks hessian dengan rumus $D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2)$. $y_i = -1$ *Class* pada atribut pertama dan $y_j = -1$ pada atribut kedua. (x_i, x_j) adalah *kernel* yang telah dihitung pada tahap sebelumnya. Inialisasi nilai $\lambda = 0.5$. Contoh perhitungan *record* pertama dan kedua adalah sebagai berikut:

$$\begin{aligned} D_{11} &= y_1 y_1 (K(x_1, x_1) + \lambda^2) \\ &= -1 \cdot -1 (0.995083 + 0.5^2) \\ &= 1 (0.995083 + 0.25) \\ &= 1.245083 \end{aligned}$$

$$\begin{aligned} D_{12} &= y_1 y_2 (K(x_1, x_2) + \lambda^2) \\ &= -1 \cdot -1 (0.731566 + 0.5^2) = 1 (0.731566 + 0.25) = 0.981566 \end{aligned}$$

Tabel 4.13 adalah hasil perhitungan *matriks hessian*.

Tabel 4.11 Hasil perhitungan *matriks hessian*

No.	A1	A2	...	A14	A15
A1	1.245083	0.981566	...	-0.99961	-0.85148

A2	0.981566	0.973099	...	-0.67836	-0.57604
...
A14	-0.99961	-0.67836	...	2.056542	1.773462
A15	-0.85148	-0.57604	...	1.773462	1.597027

Setelah menyelesaikan perhitungan matriks hessian, selanjutnya adalah menghitung E_i , $\delta\alpha_i$, dan α_i dimana ketiga fungsi tersebut dilakukan proses iterasi sesuai dengan jumlah yang dibutuhkan. Rumus untuk menghitung nilai eror adalah $E_i = \sum_{j=1}^{oj} D_{ij}$. Berikut adalah hasil dari perhitungan E_i yang dapat dilihat pada tabel 4.14 :

Tabel 4.12 Hasil perhitungan E_i

	Iterasi 1	Iterasi 2	Iterasi 3
A1	4.063324	0	4.063324
A2	3.915815	0	3.915815
A3	5.259832	0	5.259832
A4	3.541621	0	3.541621
A5	4.119581	0	4.119581
A6	4.859977	0	4.859977
A7	4.665811	0	4.665811
A8	4.655911	0	4.655911
A9	4.597434	0	4.597434
A10	4.437397	0	4.437397
A11	-0.55153	-1.10306	-1.10306
A12	-1.45619	-2.91238	-2.91238
A13	-0.61707	-1.23413	-1.23413
A14	-0.75141	-1.50282	-1.50282
A15	-0.46727	-0.93454	-0.93454

Berikut adalah contoh perhitungan nilai error data A1 pada iterasi 1:

$$\begin{aligned}
 E_i = & ((0.5 \times 1.245083) + (0.5 \times 0.981566) + \\
 & (0.5 \times 1.412589) + (0.5 \times 0.903487) + \\
 & (0.5 \times 1.197735) + (0.5 \times 1.571489) + \\
 & (0.5 \times 1.499677) + (0.5 \times 1.493692) + \\
 & (0.5 \times 1.474948) + (0.5 \times 1.453025) + \\
 & (0.5 \times (-0.89881)) + (0.5 \times (-1.28017)) + \\
 & (0.5 \times (-1.07658)) + (0.5 \times (-0.99961)) +
 \end{aligned}$$

$$(0.5 \times (-0.85148)) = 4.063324$$

Setelah mendapatkan nilai E_i , selanjutnya adalah menghitung nilai $\delta\alpha_i$ dengan rumus sebagai berikut:

Contoh perhitungan data A1 pada iterasi ke-1:

$$\begin{aligned} \delta\alpha_1 &= \min\{\text{maks}[\gamma(1 - E_1), -\alpha_1], C - \alpha_1\} \\ &= \min\{\text{maks}[0.5(1 - 4.063324), -0.5], 1 - 0.5\} \\ &= \min\{\text{maks}[0.5(-4.063324), -0.5], 0.5\} \\ &= \min\{\text{maks}[-4.063324, -0.5], 0.5\} \\ &= \min\{-4.063324, -0.5\} \\ &= -0.5 \end{aligned}$$

Tabel 4.13 Hasil perhitungan $\delta\alpha_i$

	Iterasi 1	Iterasi 2	Iterasi 3
A1	-0.5	0.5	-0.5
A2	-0.5	0.5	-0.5
A3	-0.5	0.5	-0.5
A4	-0.5	0.5	-0.5
A5	-0.5	0.5	-0.5
A6	-0.5	0.5	-0.5
A7	-0.5	0.5	-0.5
A8	-0.5	0.5	-0.5
A9	-0.5	0.5	-0.5
A10	-0.5	0.5	-0.5
A11	0.5	0	0
A12	0.5	0	0
A13	0.5	0	0
A14	0.5	0	0
A15	0.5	0	0

Setelah mendapatkan nilai $\delta\alpha_i$, selanjutnya adalah menghitung nilai α_i dengan fungsi sebagai berikut: $\alpha_1 = \alpha_1 + \delta\alpha_1$

Contoh perhitungan data A1 pada iterasi ke-1:

$$\begin{aligned} \alpha_1 &= \alpha_1 + \delta\alpha_1 \\ &= 0.5 + (-0.5) \\ &= 0 \end{aligned}$$

Tabel 4.14 Hasil perhitungan α_i

	Iterasi 1	Iterasi 2
A1	0	0.5
A2	0	0.5

A3	0	0.5
A4	0	0.5
A5	0	0.5
A6	0	0.5
A7	0	0.5
A8	0	0.5
A9	0	0.5
A10	0	0.5
A11	1	1
A12	1	1
A13	1	1
A14	1	1
A15	1	1

Setelah melakukan perhitungan sequential training svm, selanjutnya adalah mencari nilai $w \cdot x, x^+$ dan $w \cdot x, x^-$ dengan fungsi sebagai berikut :

$$w = \sum_{i=1}^n \alpha_i y_i x_i$$

$$w \cdot x^+ = \sum_{i=1}^n \alpha_i y_i x_i x^+ = \sum_{i=1}^n \alpha_i y_i k(x_i, x^+)$$

Untuk hasil perhitungan secara keseluruhan untuk mendapatkan nilai $w \cdot x^+$ dan nilai $w \cdot x^-$ dapat dilihat pada Tabel 4.17.

Tabel 4.15 Hasil perhitungan $w \cdot x^+$ dan $w \cdot x^-$

w_+	w_-	
1317.221	1577.913	-1447.57
1031.534	1285.802	-1158.67
1165.138	3473.518	-2319.33
1053.169	1004.756	-1028.96
1142.309	1710.366	-1426.34
1792.634	1325.593	-1559.11
1451.306	1361.887	-1406.6
1512.262	1062.706	-1287.48
1612.908	1144.153	-1378.53
1650.505	1478.072	-1564.29
1960.447	1317.221	3277.668
1780.626	1256.354	3036.98
1541.862	1006.422	2548.284
1716.361	1178.68	2895.04
1799.698	1240.189	3039.887

Langkah selanjutnya adalah mencari nilai b .

$$b = -\frac{1}{2}(w \cdot x^+ + w \cdot x^-)$$

$$= -110.4919$$

4.2.3 Perhitungan Manual Proses Testing SVM Dengan Kernel RBF Gaussian

Setelah melakukan perhitungan $w \cdot x, x^+$ dan $w \cdot x, x^-$ dan mengetahui hasil dari nilai bias, selanjutnya adalah menghitung data uji dengan fungsi sebagai berikut :

Tabel 4.16 Data Training 1

No.	F1	F2	F3	F4	F5	F6	F7
1	0.1	0.438292	0.447684	0.339053	0.325914	0.306607	0.526667

Tabel 4.17 Data Testing 1

No.	F1	F2	F3	F4	F5	F6	F7
1	0.309259	0.452617	0.426639	0.320118	0.337874	0.212112	0.66

$k(x, y)$, misal $x = x_1, y = x_1$ maka $k(x, y) = k(x_1, x_1) =$

$$\exp\left(-\frac{\|x_1 - x_1\|^2}{2\sigma^2}\right) = \exp\left(-\frac{0}{2 \cdot 1^2}\right) = \exp\left(-\frac{0}{2}\right) = \exp(0) = 1$$

$$(x_1 - x_1) = \begin{pmatrix} 0.1 \\ 0.438292 \\ 0.447684 \\ 0.339053 \\ 0.325914 \\ 0.306607 \\ 0.526667 \end{pmatrix} - \begin{pmatrix} 0.1 \\ 0.309259 \\ 0.452617 \\ 0.426639 \\ 0.320118 \\ 0.212112 \\ 0.66 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\|x_1 - x_1\| = \sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2} = \sqrt{0} = 0$$

$$\|x_1 - x_1\|^2 = 0^2 = 0$$

Proses selanjutnya adalah menghitung nilai $f(x)$. Kemudian menghitung fungsi klasifikasi.

$$f(x) = \sum_{i=1}^m \alpha_i y_i x_i \cdot x + b$$

$$= -7.5962$$

Fungsi Klasifikasi = $sign(-7.5962) = -1$

Setelah menghitung data uji dan fungsi klasifikasi, selanjutnya adalah membandingkan antara aktual dan hasil klasifikasi. Berikut adalah hasil perbandingan antara aktual dan hasil klasifikasi :

Tabel 4.18 Perbandingan dan Klasifikasi

Level 1	Uji1	Uji2	Uji3	Uji4	Uji5	Uji6
$\alpha yK(x,x_i)+b$	- 7.59628E-06	-7.2629E-06	8.62009E-06	6.38152E-06	2.4373E-05	2.10191E-05
f(K(x))	-1	-1	1	1	1	1
Level 2						
$\alpha yK(x,x_i)+b$	- 2.73541E-06	- 8.42873E-06	8.58943E-06	8.01112E-06	1.04722E-05	6.06128E-06
f(K(x))	-1	-1	1	1	1	1
Output Resiko :	Kualitas Rendah	Kualitas Rendah	Kualitas Tinggi	Kualitas Tinggi	Kualitas Tinggi	Kualitas Tinggi
Input Resiko :	Kualitas Rendah	Kualitas Rendah	Kualitas Standar	Kualitas Standar	Kualitas Tinggi	Kualitas Tinggi
Status Kebenaran	BENAR	BENAR	SALAH	SALAH	BENAR	BENAR

Data diatas merupakan hasil klasifikasi menggunakan metode *one against all* dari level 1 dan level 2. Perhitungan klasifikasi pada level 2 menggunakan fungsi yang sama pada level 1, dimana data uji yang digunakan menghasilkan nilai -1. Kemudian kelas y pada data latih dengan aktual “sedang” diganti menjadi 1. Sedangkan kelas y pada data latih dengan aktual “rendah” nilainya tetap -1. Seperti yang diketahui pada tabel 4.20 hasil klasifikasi pada data uji ke-1 dan ke-2 memiliki hasil -1 di level 1 yang berarti “rendah”. Pada data uji ke-5 dan ke-6 menghasilkan 1 di level 2 yang berarti “tinggi”. Selanjutnya adalah mengevaluasi perbandingan aktual dan hasil klasifikasi dengan menggunakan confusion matrix seperti pada tabel 4.21:

Tabel 4.19 Confusion Matrix

Actual Class	Predicted Class			Jumlah
	Tinggi	Sedang	Rendah	
Tinggi	2	0	0	2
Sedang	2	0	0	2
Rendah	0	0	2	2
Jumlah	4	0	2	6

Setelah mendapatkan hasil confusion matrix pada Tabel 4.21, selanjutnya adalah menghitung nilai akurasi sebagai berikut :

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

Maka nilai akurasi yang diperoleh adalah

a. $TP = C_{1,1} + C_{2,2} + C_{3,3} = 2 + 0 + 2 = 4$

b. $TN = (C_{1,1} + C_{2,2}) + (C_{1,1} + C_{3,3}) + (C_{2,2} + C_{3,3}) = 2 + 4 + 2 = 8$

c. $FP = (C_{2,1} + C_{3,1}) + (C_{1,2} + C_{3,2}) + (C_{1,3} + C_{2,3}) = 2 + 0 + 0 = 2$

d. $FN = (C_{1,2} + C_{1,3}) + (C_{2,1} + C_{2,3}) + (C_{3,1} + C_{3,2}) = 0 + 2 + 0 = 2$

$$Akurasi = \frac{4+8}{4+8+2+2} \times 100\% = 67\%$$

4.2.4 Perhitungan Manual Proses Testing SVM dengan Kernel Linier

Setelah melakukan perhitungan $w \cdot x, x^+$ dan $w \cdot x, x^-$ dan mengetahui hasil dari nilai bias, selanjutnya adalah menghitung data uji dengan fungsi sebagai berikut :

Tabel 4.20 Data Training 1

No.	F1	F2	F3	F4	F5	F6	F7
1	0.1	0.438292	0.447684	0.339053	0.325914	0.306607	0.526667

Tabel 4.21 Data Testing 1

No.	F1	F2	F3	F4	F5	F6	F7
1	0.309259	0.452617	0.426639	0.320118	0.337874	0.212112	0.66

$$K(A1, uji1) = ((0.1 * 0.309259) + (0.438292 * 0.452617) + (0.447684 * 0.426639) + (0.339053 * 0.320118) + (0.325914 * 0.337874) + (0.306607 * 0.212112) + (0.526667 * 0.66)) = 1.051594$$

Proses selanjutnya adalah menghitung nilai $f(x)$. Kemudian menghitung fungsi klasifikasi.

$$f(x) = \sum_{i=1}^m \alpha_i y_i x_i \cdot x + b$$

$$= -1.512629356$$

Fungsi Klasifikasi = $sign(-1.512629356) = -1$

Setelah menghitung data uji dan fungsi klasifikasi, selanjutnya adalah membandingkan antara aktual dan hasil klasifikasi. Tabel 4.24 adalah hasil perbandingan antara aktual dan hasil klasifikasi

Tabel 4.22 Perbandingan dan Klasifikasi

Level 1	Uji1	Uji2	Uji3	Uji4	Uji5	Uji6
$\alpha y K(x, x_i) + b$	0.303852 358	0.290517 469	0.344806 616	0.255262 822	0.97492 8969	0.84076 9967
$f(K(x))$	-1	-1	1	1	1	1
Level 2						
$\alpha y K(x, x_i) + b$	0.109417 312	0.337150 698	0.343578 494	0.320445 555	0.41889 3651	0.24245 4972
$f(K(x))$	-1	-1	1	1	1	1
Output Resiko :	Kualitas Rendah	Kualitas Rendah	Kualitas Tinggi	Kualitas Tinggi	Kualitas Tinggi	Kualitas Tinggi
Input Resiko :	Kualitas Rendah	Kualitas Rendah	Kualitas Standar	Kualitas Standar	Kualitas Tinggi	Kualitas Tinggi
Status Kebenaran	BENAR	BENAR	SALAH	SALAH	BENAR	BENAR

Data diatas merupakan hasil klasifikasi menggunakan metode *one against all* dari level 1 dan level 2. Perhitungan klasifikasi pada level 2 menggunakan fungsi yang sama pada level 1, dimana data uji yang digunakan menghasilkan nilai = -1. Kemudian kelas y pada data latih dengan aktual "sedang" diganti menjadi 1. Sedangkan kelas y pada data latih dengan aktual "rendah" nilainya tetap -1. Seperti yang diketahui pada tabel 4.24 hasil klasifikasi pada data uji ke-1 dan ke-2 memiliki hasil -1 di level 1 yang berarti "rendah". Pada data uji ke-5 dan ke-6 menghasilkan 1 di level 2 yang berarti "tinggi". Selanjutnya adalah mengevaluasi perbandingan aktual dan hasil klasifikasi dengan menggunakan confusion matrix seperti pada tabel 4.25.

Tabel 4.23 Confusion Matrix

Actual Class	Predicted Class			Jumlah
	Tinggi	Sedang	Rendah	
Tinggi	2	0	0	2
Sedang	2	0	0	2
Rendah	0	0	2	2
Jumlah	4	0	2	6

Setelah mendapatkan hasil confusion matrix pada Tabel 4.25, selanjutnya adalah menghitung nilai akurasi sebagai berikut :

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

Maka nilai akurasi yang diperoleh adalah

$$a. TP = C_{1,1} + C_{2,2} + C_{3,3} = 2 + 0 + 2 = 4$$

$$b. TN = (C_{1,1} + C_{2,2}) + (C_{1,1} + C_{3,3}) + (C_{2,2} + C_{3,3}) = 2 + 4 + 2 = 8$$

$$c. FP = (C_{2,1} + C_{3,1}) + (C_{1,2} + C_{3,2}) + (C_{1,3} + C_{2,3}) = 2 + 0 + 0 = 2$$

$$d. FN = (C_{1,2} + C_{1,3}) + (C_{2,1} + C_{2,3}) + (C_{3,1} + C_{3,2}) = 0 + 2 + 0 = 2$$

$$Akurasi = \frac{4 + 8}{4 + 8 + 2 + 2} \times 100\% = 67\%$$

4.3 Perancangan Pengujian Sistem

Pada tahap perancangan pengujian sistem ini dilakukan untuk menguji validasi dari perangkat lunak yang telah dibuat. Sistem yang baik adalah sistem yang memiliki sedikit error pada saat digunakan. Proses pengujian sistem ini meliputi:

1. Pengujian Berdasarkan *K-fold* Data.
2. Pengujian Jenis *Kernel*.
3. Pengujian Berdasarkan Nilai λ (*Lambda*).
4. Pengujian Berdasarkan Nilai γ (*Gamma*).
5. Pengujian Berdasarkan Nilai *C* (*Complexity*).
6. Pengujian Berdasarkan σ *Kernel* RBF
7. Pengujian Berdasarkan Nilai *Itermax*

4.4 Evaluasi Hasil

Hasil Evaluasi ini digunakan untuk mengetahui bagaimana hasil kinerja dari sistem dan kemudian dianalisis, Tahapan ini juga berisi kesimpulan yang memiliki tujuan untuk pengembangan sistem agar menjadi lebih baik dan mampu memberikan saran terkait dengan kekurangan pada penelitian ini.