

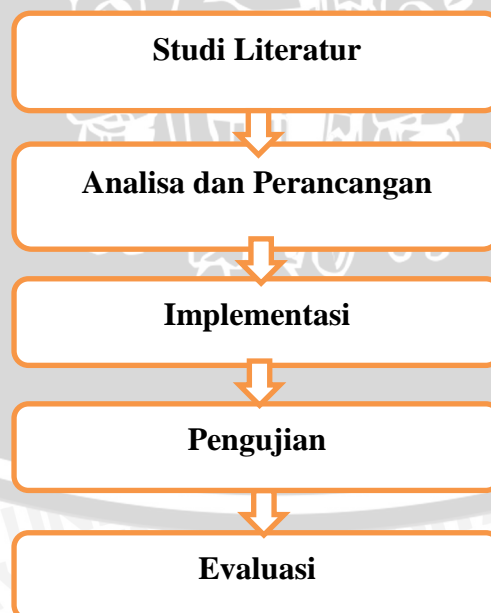
BAB III

METODOLOGI PENELITIAN

Pada bab ini akan dibahas mengenai metode perancangan yang digunakan dan langkah – langkah yang dilakukan dalam penelitian untuk melakukan klasifikasi ada atau tidaknya virus hepatitis pada pasien dengan menggunakan metode *Support Vector Machine* (SVM). Tahapan yang dilakukan dalam penelitian ini yaitu sebagai berikut :

1. Mempelajari literatur yang berhubungan metode *Support Vector Machine* (SVM) dan penyakit hepatitis.
2. Melakukan analisa dan perancangan sistem dengan metode yang digunakan.
3. Membuat perangkat lunak berdasarkan analisa dan perancangan yang dilakukan.
4. Melakukan uji coba klasifikasi penyakit hepatitis menggunakan perangkat lunak yang telah dibuat.
5. Melakukan evaluasi output hasil uji coba.

Langkah – langkah yang dilakukan dapat dilihat pada gambar 3.1 berikut :



Gambar 3.1 Diagram Alir Penelitian

3.1 Studi Literatur

Studi literatur dalam penelitian ini dimaksudkan untuk mendukung penyelesaian masalah dan merealisasikan tujuan. Studi literatur dilakukan dengan mencari sumber berupa jurnal, buku – buku referensi, skripsi serta hasil browsing dari internet untuk mendapatkan referensi terkait dengan dasar penelitian yaitu teori tentang metode *Support Vector Machine* dan penyakit hepatitis. Setelah data didapatkan, kemudian dilakukan analisa agar kemudian dapat diimplementasikan ke dalam program yang dibuat.

3.2 Analisa Data

Dataset yang digunakan adalah dataset *Hepatitis Domain* yang diperoleh dari *University California Irvine* (UCI). Dataset tersebut dapat diperoleh dari situs *UCI Machine Learning Repository*. Pada dataset tersebut terdiri dari 20 atribut (termasuk kelas), informasi mengenai dataset ini dapat dilihat pada tabel 2.1. *Dataset* ini terdiri dari 155 data, dimana 32 data merupakan kelas *die* dan 123 data merupakan kelas *live*, dimana terdapat 5 atribut dengan data *continue* (*real*) dan 15 atribut merupakan data *categorical*. Dalam *dataset Hepatitis Domain* juga terdapat data dengan *missing value*. Untuk informasi mengenai *missing value* dari dataset ini dapat dilihat pada tabel 3.1 berikut ini :

Tabel 3.1 Informasi *Missing Value* dataset *Hepatitis Domain*

Attribut	Jumlah data Missing Value
1	0
2	0
3	0
4	1
5	0
6	1
7	1
8	1
9	10
10	11
11	5

12	5
13	5
14	5
15	6
16	29
17	4
18	16
19	67
20	0

Karena sistem tidak menangani masalah *missing value*, maka nilai atribut data yang terdapat *missing value* akan diberi nilai 0, dan kemudian akan dilatih dengan metode *Support Vector Machine* (SVM).

3.3 Analisa Sistem

3.3.1 Deskripsi Sistem

Pada sistem yang dibuat ini merupakan perangkat lunak yang mengimplementasikan metode *Support Vector Machine* (SVM) untuk melakukan klasifikasi terhadap penyakit hepatitis. Pada perangkat lunak ini akan melatih dataset *Hepatitis Domain* dengan terlebih dahulu melakukan normalisasi untuk membuat nilai data lebih kecil, sehingga dapat mengurangi waktu komputasi. Selanjutnya akan dilakukan proses *training* dengan metode *Support Vector Machine* (SVM) untuk mencari bidang pemisah dari kelas sesuai pada dataset *Hepatitis Domain*. Kemudian dalam proses pelatihan dengan menggunakan metode ini dataset *Hepatitis Domain* akan dibagi dua bagian yaitu sebagai data latih dan sebagai data uji.

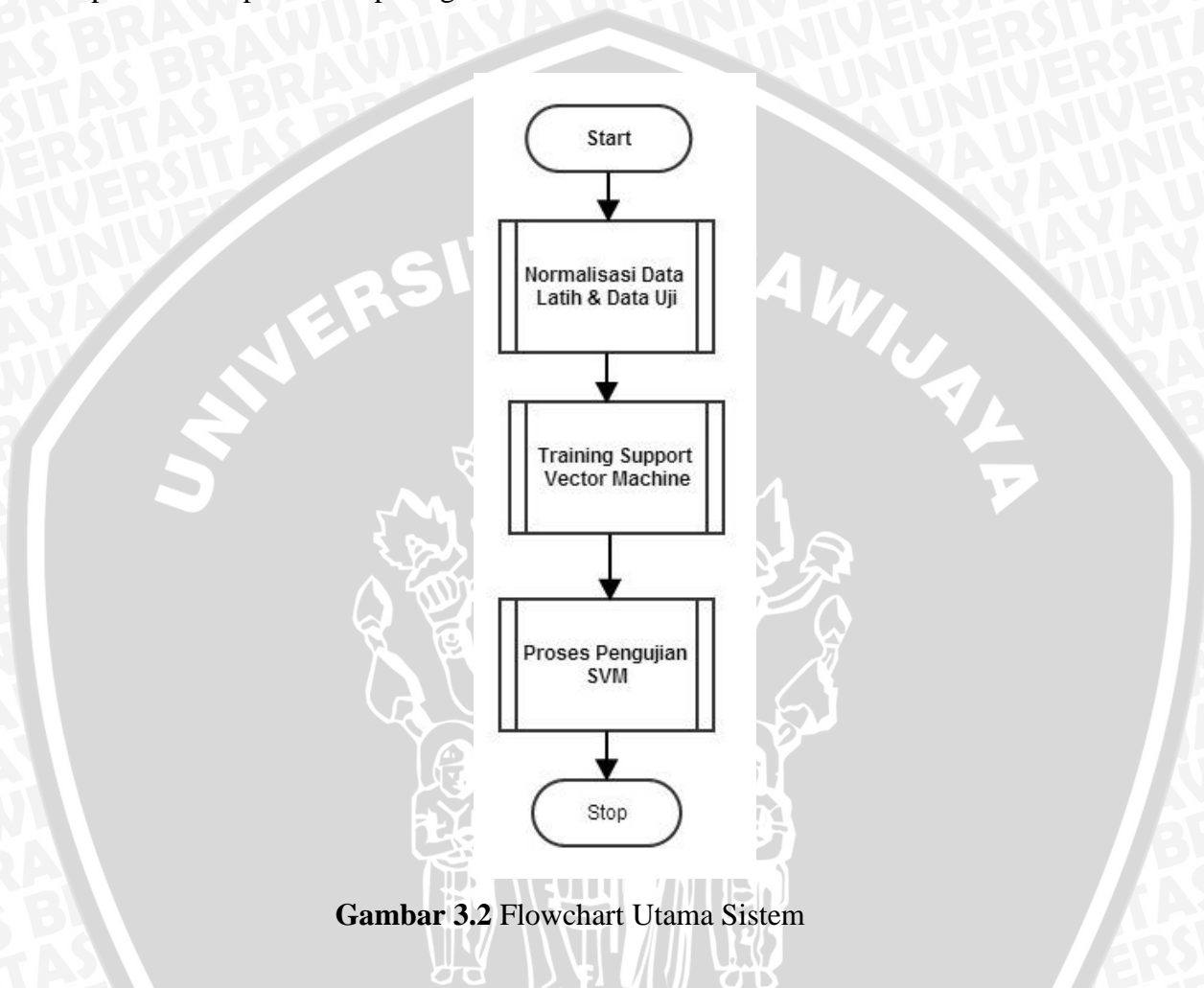
3.3.2 Batasan Sistem

Sistem yang dibuat memiliki batasan sebagai berikut :

1. Sistem tidak menangani *missing value* secara khusus
2. Data yang digunakan adalah data dengan jumlah kelas = 2
3. Metode yang digunakan adalah *Support Vector Machine* (SVM)

3.3.3 Alur Perancangan Sistem

Adapun alur perancangan sistem klasifikasi kondisi penderita penyakit hepatitis ini dapat dilihat pada gambar 3.1



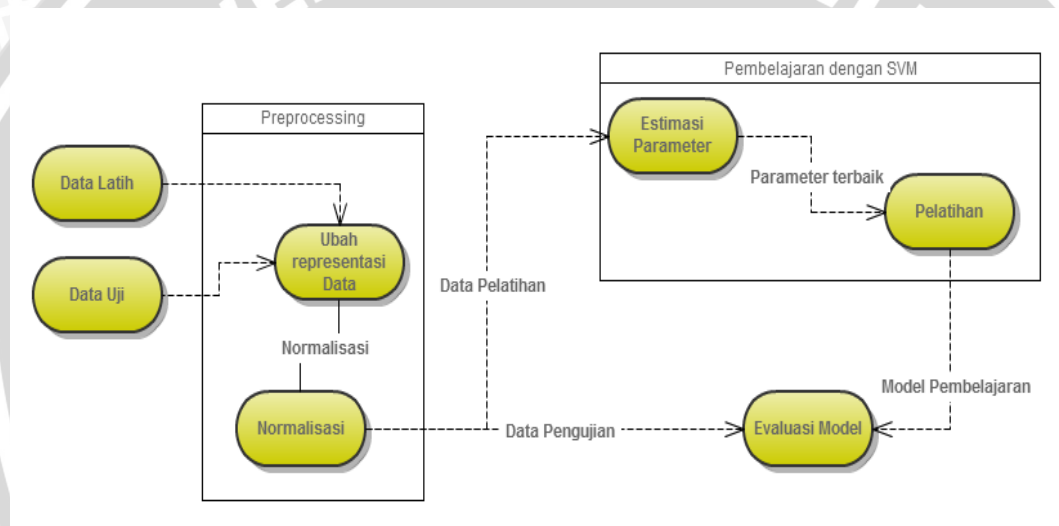
Gambar 3.2 Flowchart Utama Sistem

Pada gambar 3.1 merupakan *flowchart* utama dari sistem klasifikasi kondisi penderita penyakit hepatitis dengan menggunakan metode SVM, dimana terdapat 3 proses utama dalam sistem yaitu proses Normalisasi data latih dan data uji, Proses Training dengan SVM, dan Proses Pengujian dengan SVM. Langkah kerjanya adalah, setelah data latih dan data uji dipilih dari dataset, maka nilai – nilai atribut data akan di normalisasi, hasil dari normalisasi ini kemudian akan dilatih dengan menggunakan metode SVM untuk mendapatkan parameter – parameter SVM. Setelah didapatkan nilai parameter tersebut maka dapat dilakukan pengujian dengan menggunakan data uji, untuk mendapatkan nilai

klasifikasi pada data uji, dimana pada proses uji ini juga akan dilakukan evaluasi dengan mencari nilai akurasi, *sensitivity*, dan *spesitivity*.

3.4 Perancangan Sistem

Pada subbab ini akan dijelaskan mengenai tahapan yang akan dilakukan agar dapat membentuk sistem yang akan diharapkan dengan menggunakan metode *Support Vector Machine*. Dalam proses klasifikasi penyakit hepatitis ini memerlukan inputan berupa dataset *Hepatitis Domain* dan menghasilkan fungsi pemisah sesuai kelas pada dataset tersebut. Proses *training* dengan *Support Vector Machine* dapat dilihat pada gambar 3.3 berikut :



Gambar 3.3 Pembelajaran dengan SVM

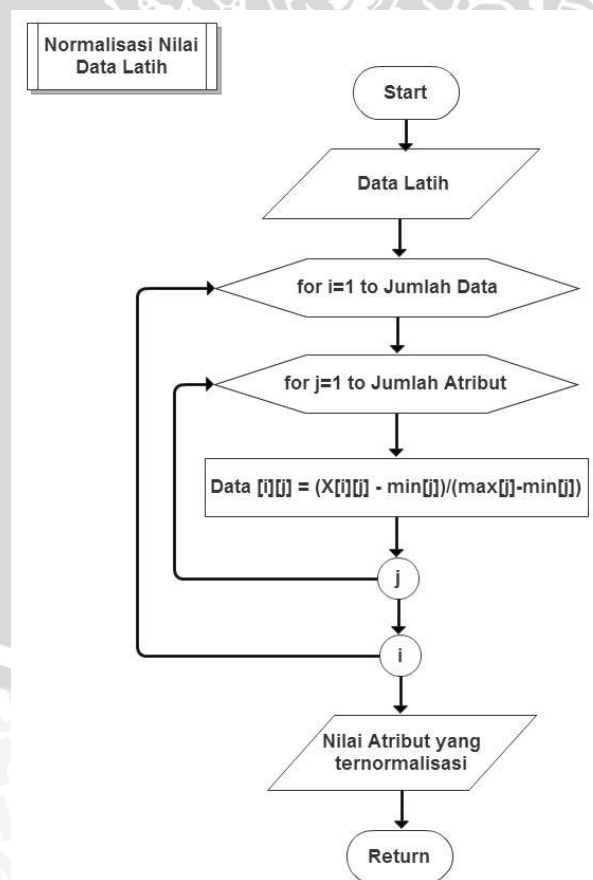
Berdasarkan gambar 3.3 adapun proses pembelajaran dengan metode Support Vector Machine yaitu :

1. Proses *input* data yaitu data *latih* dan *data uji*, nilai w (*weight*), nilai b (*bias*), serta nilai *slack variable* Proses Normalisasi data, yaitu dengan merubah skala data asli menjadi skala 0-1.
2. Proses pelatihan data yaitu data *latih* untuk dibuat model, dimana akan menghasilkan parameter berupa nilai w (*weight*), nilai b (*bias*), dan *slack variable* melalui proses *Quadratic Programming* dengan bantuan *Solver*.
3. Proses Evaluasi Model yaitu melakukan proses pengujian dengan menggunakan data uji yang dipilih untuk proses klasifikasi.

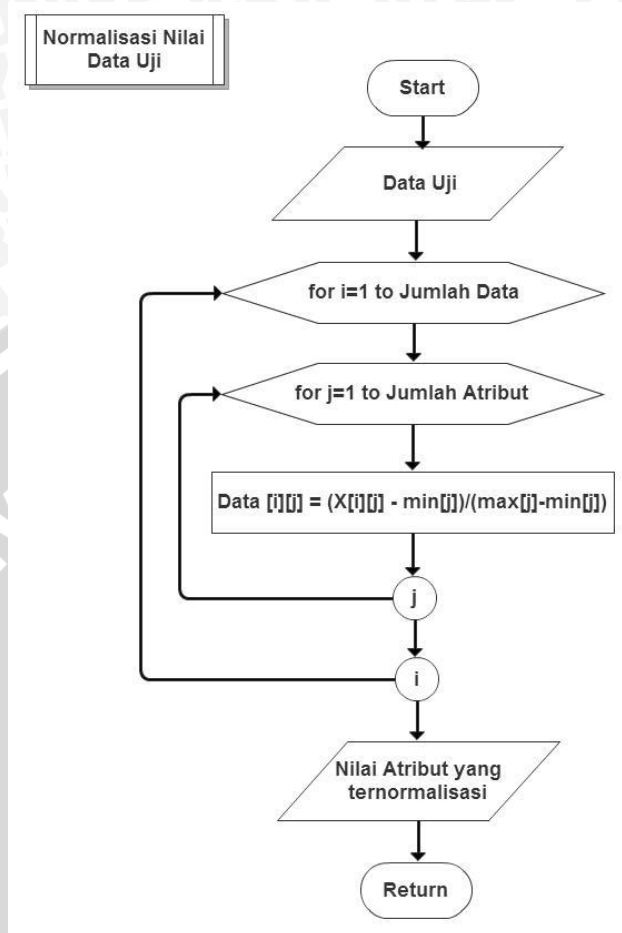
3.4.1 Normalisasi Nilai Atribut

Langkah – langkah yang dilakukan dalam proses normalisasi atribut data yaitu sebagai berikut :

1. Nilai *input* untuk data uji dan data latih.
 2. Melakukan iterasi untuk mencari nilai minimum dan nilai maksimum untuk tiap atribut.
 3. Masukkan nilai maksimum dan minimum dengan menggunakan persamaan 2.3 untuk melakukan *scaling* dengan Batas Atas (BA) =1 dan Batas Bawah (BB) = 0.
 4. *Output* dari proses perhitungan adalah nilai atribut yang ternormalisasi
- Adapun *Flowchart* untuk proses normalisasi atribut ini dapat ditunjukkan pada gambar 3.4 dan gambar 3.5



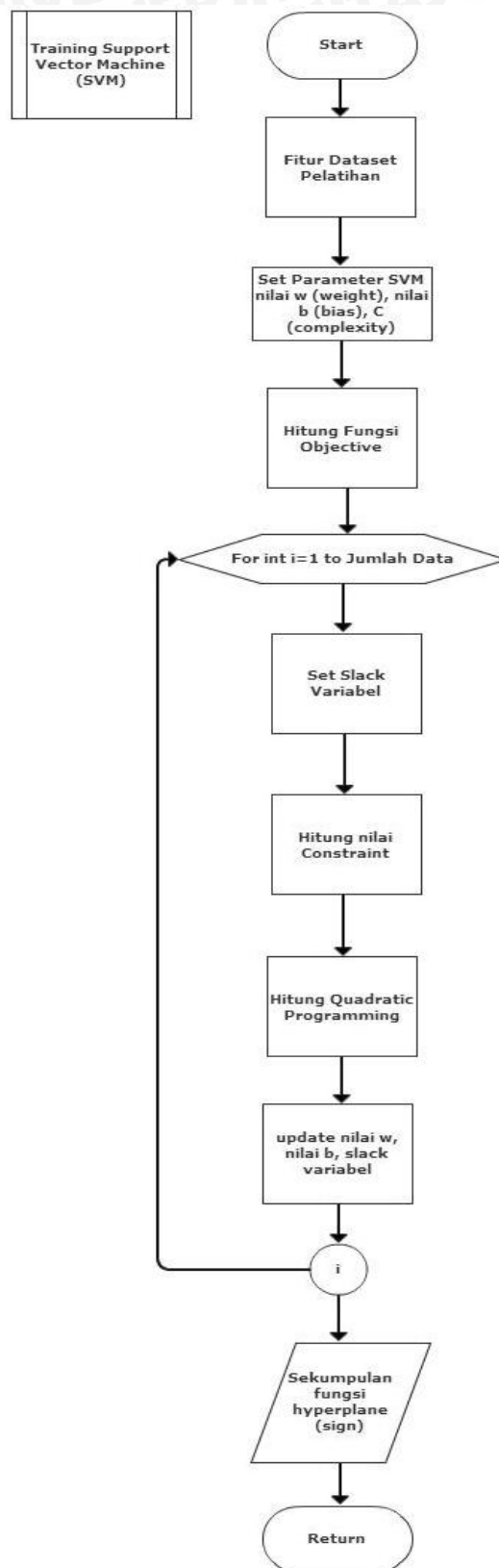
Gambar 3.4 Flowchart Proses Normalisasi Data Latih



Gambar 3.5 Flowchart Proses Normalisasi Data Uji

3.4.2 Pelatihan dengan *Support Vector Machine*

Pada proses ini akan dijelaskan mengenai tahapan yang dilakukan dalam pada proses *training* dengan metode *Support Vector Machine* (SVM). Setiap proses *learning* merupakan pengklasifikasian biner dari sekumpulan objek yang ada. Fungsi pemisah atau *hyperplane* pada *Support Vector Machine* (SVM) merupakan fungsi dengan nilai margin yang maksimal. Margin adalah jarak antara *hyperplane* dengan *support vector* masing-masing objek yang di training. Pada fungsi SVM terdapat parameter *Support Vector Machine* yaitu nilai w (weight), nilai b (bias), C (*complexity*) yang digunakan untuk proses *learning*, dimana nilai tersebut berpengaruh pada presisi pengklasifikasian. Adapun tahapan proses *training* pada SVM adalah sebagai berikut :



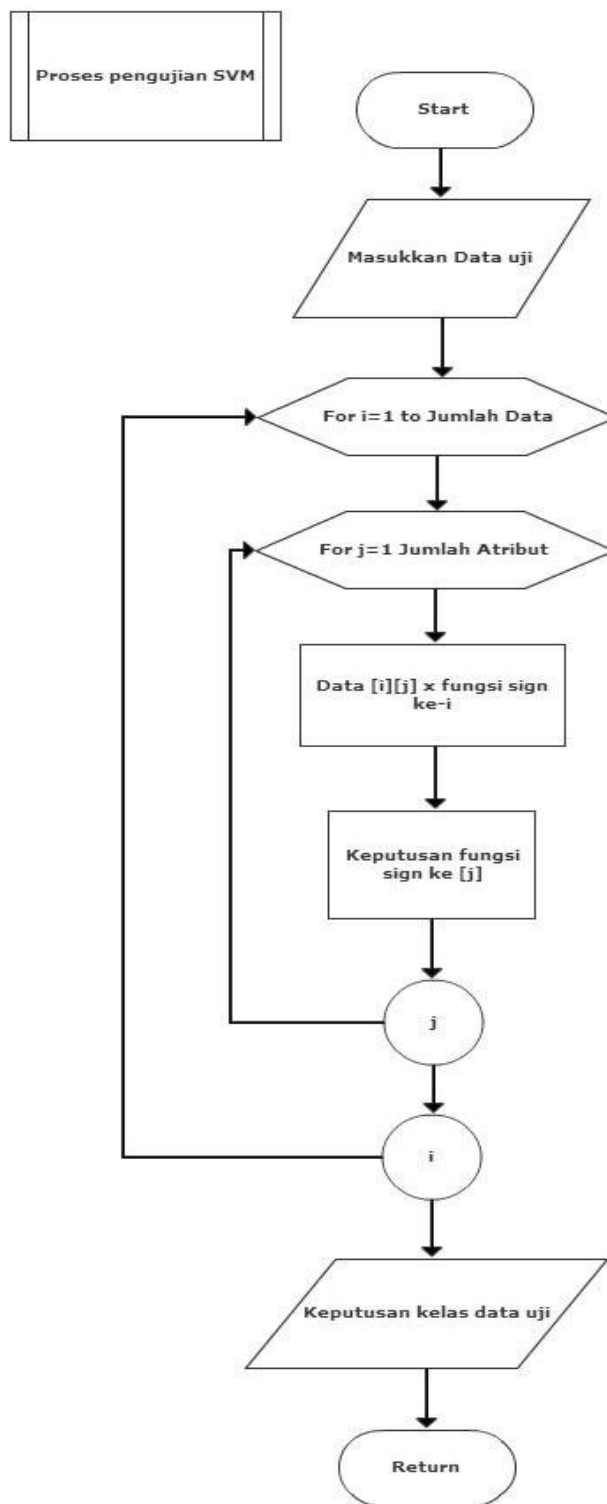
Gambar 3.6 Proses Training Support Vector Machine

Berikut adalah penjelasan tahapan dari proses pembelajaran metode *Support Vector Machine* (SVM) :

1. Langkah awal adalah memasukkan data latih yang sebelumnya telah dilakukan proses normalisasi data.
2. Selanjutnya melakukan *set* parameter SVM, dimana data kemudian akan dipetakan ke *dot product kernel* SVM, dimana kernel yang digunakan adalah *kernel linear*. Untuk parameter yang akan dipakai yaitu *set* nilai w (*weight*), nilai b (*bias*), *slack variable*, dan nilai C (*complexity*).
3. Nilai *slack variable* akan diset sejumlah record pada dataset, maka perhitungannya dilakukan looping sejumlah n - *record* dataset. Tujuan menggunakan *variable slack* adalah untuk mengatasi kondisi ketidaklayakan (*infeasibility*) dari pembatas (*constraint*).
4. Setelah dilakukan perhitungan nilai *constraint*, nilai *constraint* nantinya akan menjadi nilai batas pada proses perhitungan *Quadratic Programming*
5. Pada tahapan ini adalah proses perhitungan *Quadratic Programming* untuk mendapatkan nilai w (*weight*) sebanyak atribut dari dataset, dan nilai b (*bias*).
6. Nilai w (*weight*), nilai b (*bias*) dan nilai *slack variable* telah diperoleh, kemudian nilai tersebut akan diupdate untuk di proses untuk mendapatkan nilai *constraint* yang baru untuk mendapatkan fungsi *hyperplane* (pemisah) yang optimal.
7. Hasil akhir dari training SVM adalah sekumpulan fungsi *hyperplane* yang kemudian akan digunakan dalam proses *testing*.

3.4.3 Proses Pengujian dengan Support Vector Machine

Proses ini merupakan tahapan pengujian data untuk proses klasifikasi dengan metode SVM. Setelah data dilatih pada proses pelatihan, maka akan didapatkan model SVM. Data uji diproses dengan menggunakan model SVM, dimana terdapat parameter – parameter yang telah didapatkan dari proses pelatihan sebelumnya yang akan digunakan untuk proses pengujian dalam klasifikasi penyakit Hepatitis. Adapun tahapan pengujian dapat dilihat pada gambar 3.7



Gambar 3.7 Proses Pengujian dengan SVM

Berikut ini adalah penjelasan tahapan proses pengujian SVM sesuai dengan gambar 3.7 untuk menentukan objek klasifikasi :

1. Proses pengujian dilakukan dengan pada data uji yang akan diuji dengan fungsi (*hyperplane*).
2. Data uji yang telah dinormalisasi dan telah dipetakan dengan fungsi kernel linear akan diproses dengan parameter – parameter SVM dan fungsi *hyperplane* (*sign*).
3. Setelah data diproses melalui fungsi *sign* kemudian akan didapatkan nilai klasifikasi dari dataset.
4. Dari dataset yang telah diproses melalui pengujian melalui fungsi *sign*, akan didapatkan objek klasifikasi dari setiap data uji. Sehingga kemudian didapatkan nilai klasifikasi dari data uji yang ada.

3.5 Perhitungan Manual

Perhitungan manual menggunakan data *sample Hepatitis Domain* dimana dataset terdiri dari 19 atribut dan 1 kelas. Informasi mengenai dataset Hepatitis Domain dapat dilihat pada tabel 2.1 pada bab sebelumnya. Adapun data yang bersifat *categorical* maka akan dikonversi nilainya ke bentuk data *continue* untuk mempermudah proses perhitungan. Pada fitur jenis kelamin, data berupa laki – laki dan perempuan, untuk laki – laki (*male*) diubah menjadi nilai 1, sedangkan perempuan (*female*) diubah menjadi 2. Kemudian untuk data yang bernilai *yes/no* maka akan diubah menjadi 1 untuk nilai *yes* dan 2 untuk nilai *noi*. Sedangkan kelas ada 2 yaitu kelas *live* dan kelas *die*. Kelas hepatitis akut (*live*) diberi nilai 1 dan kelas hepatitis kronis (*die*) diberi nilai -1. Pada perhitungan manual data yang akan digunakan sebagai data latih sebanyak 20 *record*, sedangkan untuk data uji sebanyak 8 *record*. Adapun *sample* untuk data latih dan data uji dapat dilihat pada tabel 3.2.

Tabel 3.2 Data Latih dan Data Uji

NO	Atr 1	Atr 2	Atr 3	Atr 4	Atr 5	Atr 6	Atr 7	Atr 8	Atr 9	Atr 10	Atr 11	Atr 12	Atr 13	Atr 14	Atr 15	Atr 16	Atr 17	Atr 18	Atr 19	Class
1	51	2	2	2	1	2	1	1	2	1	1	2	2	0	0	0	0	0	2	-1
2	39	2	2	1	1	1	2	1	1	2	2	2	2	2.3	280	98	3.8	40	2	-1
3	62	2	2	2	1	1	2	0	0	2	2	2	2	1	0	60	0	0	2	-1
4	37	2	1	2	1	2	2	1	2	2	1	2	2	0.6	67	28	4.2	0	2	-1
5	57	2	1	2	1	1	1	1	2	2	1	1	2	4.1	0	48	2.6	73	2	-1
6	34	2	2	2	1	1	2	2	1	2	1	2	2	2.8	127	182	0	0	2	-1
7	58	2	1	2	1	2	2	2	1	1	1	2	2	2	167	242	3.3	0	2	-1
8	44	2	2	2	1	1	2	1	2	1	2	2	1	0.9	135	55	0	41	1	-1
9	30	2	1	2	1	1	1	1	1	2	1	1	1	2.5	165	64	2.8	0	1	-1
10	54	2	2	2	1	1	2	0	0	1	2	1	2	3.9	120	28	3.5	43	1	-1
11	38	2	2	1	2	2	2	2	1	2	2	2	2	0.7	70	28	4.2	62	2	1
12	22	1	1	1	1	2	2	1	2	2	2	2	2	0.9	48	20	4.2	64	2	1
13	27	2	1	2	1	1	1	2	1	1	1	2	2	1.2	133	98	4.1	39	2	1
14	31	2	1	2	2	2	2	1	2	2	2	2	2	1	85	20	4	100	2	1
15	42	2	1	2	2	2	2	1	2	2	2	2	2	0.9	60	63	4.7	47	2	1
16	25	1	2	1	2	2	2	1	2	2	2	2	2	0.4	45	18	4.3	70	2	1
17	27	2	2	2	1	1	2	1	2	2	2	2	2	0.8	95	46	3.8	100	2	1
18	49	2	2	1	1	1	1	1	1	2	1	2	2	0.6	85	48	3.7	0	2	1
19	58	1	1	2	1	2	2	1	1	2	1	2	2	1.4	175	55	2.7	36	2	1
20	61	2	2	2	1	2	2	2	1	2	2	2	2	1.3	78	25	3.8	100	2	1
21	61	2	2	2	1	1	2	0	0	2	1	2	2	0	0	0	0	0	1	?
22	56	2	2	2	1	1	1	2	1	2	1	2	2	2.9	90	153	4	0	1	?
23	50	2	1	2	1	2	2	1	1	1	2	1	1	2.8	155	75	2.4	32	1	?
24	57	2	2	2	1	1	2	1	2	2	1	1	2	4.6	82	55	3.3	30	1	?
25	40	2	2	2	1	2	2	1	1	2	2	2	2	0.6	62	166	4	63	2	?
26	38	2	1	2	2	2	2	1	2	2	2	2	2	0.7	53	42	4.1	85	1	?
27	38	2	2	1	2	2	2	2	1	2	2	2	2	0.7	70	28	4.2	62	2	?
28	22	1	1	1	1	2	2	1	2	2	2	2	2	0.9	48	20	4.2	64	2	?

Tahap 1 : Melakukan Normalisasi terhadap nilai pada setiap atribut

Pada tahapan ini merupakan proses normalisasi pada data uji dan data latih dimana setiap nilai dari atribut pada dataset akan dinormalisasi sesuai skala yang ditentukan. Proses normalisasi terlebih dahulu mencari nilai *max* dan nilai *min* dari setiap atribut yang ada. Adapun contoh perhitungan dari normalisasi nilai *record* pertama pada atribut pertama adalah :

- $X = 51$ (merupakan nilai *record* pertama pada atribut-1)
- $X_{\min} = 22$ (nilai minimum pada atribut ke-1)
- $X_{\max} = 62$ (nilai maksimum pada atribut ke-1)

Setelah diperoleh nilai tersebut kemudian dilakukan proses perhitungan dengan menggunakan persamaan (2.3).

$$\hat{X} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \times (1 - 0) + (0) = \frac{51 - 22}{62 - 22} \times 1 = 0.725$$

Dari hasil perhitungan diperoleh bahwa nilai normalisasi untuk atribut ke-1 *record* ke-1 adalah 0.725. Selanjutnya proses yang sama akan dilakukan pada nilai *record* pada seluruh atribut. Hasil dari proses ini dapat dilihat pada Tabel 3.3 dan 3.4

Tabel 3.3 Data latih setelah di normalisasi

Atr 1	Atr 2	Atr 3	Atr 4	Atr 5	Atr 6	Atr 7	Atr 8	Atr 9	Atr 10	Atr 11	Atr 12	Atr 13	Atr 14	Atr 15	Atr 16	Atr 17	Atr 18	Atr 19
0.73	1.00	1.00	1.00	0.00	1.00	0.00	0.50	1.00	0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00
0.43	1.00	1.00	0.00	0.00	0.00	1.00	0.50	0.50	1.00	1.00	1.00	1.00	0.56	1.00	0.40	0.81	0.40	1.00
1.00	1.00	1.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	1.00	1.00	1.00	0.24	0.00	0.25	0.00	0.00	1.00
0.38	1.00	0.00	1.00	0.00	1.00	1.00	0.50	1.00	1.00	0.00	1.00	1.00	0.15	0.24	0.12	0.89	0.00	1.00
0.88	1.00	0.00	1.00	0.00	0.00	0.00	0.50	1.00	1.00	0.00	0.00	1.00	1.00	0.00	0.20	0.55	0.73	1.00
0.30	1.00	1.00	1.00	0.00	0.00	1.00	1.00	0.50	1.00	0.00	1.00	1.00	0.68	0.45	0.75	0.00	0.00	1.00
0.90	1.00	0.00	1.00	0.00	1.00	1.00	1.00	0.50	0.00	0.00	1.00	1.00	0.49	0.60	1.00	0.70	0.00	1.00
0.55	1.00	1.00	1.00	0.00	0.00	1.00	0.50	1.00	0.00	1.00	1.00	0.00	0.22	0.48	0.23	0.00	0.41	0.00
0.20	1.00	0.00	1.00	0.00	0.00	0.00	0.50	0.50	1.00	0.00	0.00	0.00	0.61	0.59	0.26	0.60	0.00	0.00
0.80	1.00	1.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	1.00	0.95	0.43	0.12	0.74	0.43	0.00
0.40	1.00	1.00	0.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	0.17	0.25	0.12	0.89	0.62	1.00
0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	0.22	0.17	0.08	0.89	0.64	1.00
0.13	1.00	0.00	1.00	0.00	0.00	0.00	1.00	0.50	0.00	0.00	1.00	1.00	0.29	0.48	0.40	0.87	0.39	1.00
0.23	1.00	0.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	0.24	0.30	0.08	0.85	1.00	1.00
0.50	1.00	0.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	0.22	0.21	0.26	1.00	0.47	1.00
0.08	0.00	1.00	0.00	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	0.10	0.16	0.07	0.91	0.70	1.00
0.13	1.00	1.00	1.00	0.00	0.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	0.20	0.34	0.19	0.81	1.00	1.00
0.68	1.00	1.00	0.00	0.00	0.00	0.00	0.50	0.50	1.00	0.00	1.00	1.00	0.15	0.30	0.20	0.79	0.00	1.00
0.90	0.00	0.00	1.00	0.00	1.00	1.00	0.50	0.50	1.00	0.00	1.00	1.00	0.34	0.63	0.23	0.57	0.36	1.00
0.98	1.00	1.00	1.00	0.00	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	0.32	0.28	0.10	0.81	1.00	1.00

Tabel 3.4 Data uji setelah di normalisasi

Atr 1	Atr 2	Atr 3	Atr 4	Atr 5	Atr 6	Atr 7	Atr 8	Atr 9	Atr 10	Atr 11	Atr 12	Atr 13	Atr 14	Atr 15	Atr 16	Atr 17	Atr 18	Atr 19
1.00	1.00	1.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
0.87	1.00	1.00	1.00	0.00	0.00	0.00	1.00	0.50	1.00	0.00	1.00	1.00	0.63	0.58	0.92	0.95	0.00	0.00
0.72	1.00	0.00	1.00	0.00	1.00	1.00	0.50	0.50	0.00	1.00	0.00	0.00	0.61	1.00	0.45	0.57	0.38	0.00
0.90	1.00	1.00	1.00	0.00	0.00	1.00	0.50	1.00	1.00	0.00	0.00	1.00	1.00	0.53	0.33	0.79	0.35	0.00
0.46	1.00	1.00	1.00	0.00	1.00	1.00	0.50	0.50	1.00	1.00	1.00	1.00	0.13	0.40	1.00	0.95	0.74	1.00
0.41	1.00	0.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	0.15	0.34	0.25	0.98	1.00	0.00
0.41	1.00	1.00	0.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	0.15	0.45	0.17	1.00	0.73	1.00
0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	0.20	0.31	0.12	1.00	0.75	1.00

Tahap 2 : Proses Training dengan metode SVM

Setelah data latih di normalisasi maka kemudian data akan dipetakan ke kernel SVM yaitu kernel linear, dengan mengimplementasikan persamaan (2.19) berikut.

$$K(x_i, x) = x_i^T x$$

Karena, setiap nilai atribut dianggap 1 dimensional data, maka untuk proses perhitungan kernel linear menjadi :

$$\begin{aligned} K(x_i, x) &= (x \cdot x) \\ &= x^2 \end{aligned}$$

Maka : x_1 = nilai atribut ke-1

$$K(x_1, x) = 0.725^2 = 0.526$$

$$K(x_1, x) = 0.425^2 = 0,181$$

$$K(x_1, x) = 1^2 = 1$$

.... dan seterusnya.

$$K(x_1, x) = 0.975^2 = 0.951$$

Perhitungan nilai pada atribut kedua, dilakukan sama seperti perhitungan nilai pada atribut yang pertama. Adapun hasil perhitungan kernel data latih serta data uji dapat dilihat pada tabel 3.5 dan tabel 3.6 .

Tabel 3.5 Data latih hasil perhitungan Kernel

Atr 1	Atr 2	Atr 3	Atr 4	Atr 5	Atr 6	Atr 7	Atr 8	Atr 9	Atr 10	Atr 11	Atr 12	Atr 13	Atr 14	Atr 15	Atr 16	Atr 17	Atr 18	Atr 19
0.53	1.00	1.00	1.00	0.00	1.00	0.00	0.25	1.00	0.00	0.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	1.00
0.18	1.00	1.00	0.00	0.00	0.00	1.00	0.25	0.25	1.00	1.00	1.00	1.00	0.31	1.00	0.16	0.65	0.16	1.00
1.00	1.00	1.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	1.00	1.00	1.00	0.06	0.00	0.06	0.00	0.00	1.00
0.14	1.00	0.00	1.00	0.00	1.00	1.00	0.25	1.00	1.00	0.00	1.00	1.00	0.02	0.06	0.01	0.80	0.00	1.00
0.77	1.00	0.00	1.00	0.00	0.00	0.00	0.25	1.00	1.00	0.00	0.00	1.00	1.00	0.00	0.04	0.31	0.53	1.00
0.09	1.00	1.00	1.00	0.00	0.00	1.00	1.00	0.25	1.00	0.00	1.00	1.00	0.47	0.21	0.57	0.00	0.00	1.00
0.81	1.00	0.00	1.00	0.00	1.00	1.00	1.00	0.25	0.00	0.00	1.00	1.00	0.24	0.36	1.00	0.49	0.00	1.00
0.30	1.00	1.00	1.00	0.00	0.00	1.00	0.25	1.00	0.00	1.00	1.00	0.00	0.05	0.23	0.05	0.00	0.17	0.00
0.04	1.00	0.00	1.00	0.00	0.00	0.00	0.25	0.25	1.00	0.00	0.00	0.00	0.37	0.35	0.07	0.35	0.00	0.00
0.64	1.00	1.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	1.00	0.90	0.18	0.01	0.55	0.18	0.00
0.16	1.00	1.00	0.00	1.00	1.00	1.00	1.00	0.25	1.00	1.00	1.00	1.00	0.03	0.06	0.01	0.80	0.38	1.00
0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.25	1.00	1.00	1.00	1.00	1.00	0.05	0.03	0.01	0.80	0.41	1.00
0.02	1.00	0.00	1.00	0.00	0.00	0.00	1.00	0.25	0.00	0.00	1.00	1.00	0.09	0.23	0.16	0.76	0.15	1.00
0.05	1.00	0.00	1.00	1.00	1.00	1.00	0.25	1.00	1.00	1.00	1.00	1.00	0.06	0.09	0.01	0.72	1.00	1.00
0.25	1.00	0.00	1.00	1.00	1.00	1.00	0.25	1.00	1.00	1.00	1.00	1.00	0.05	0.05	0.07	1.00	0.22	1.00
0.01	0.00	1.00	0.00	1.00	1.00	1.00	0.25	1.00	1.00	1.00	1.00	1.00	0.01	0.03	0.01	0.84	0.49	1.00
0.02	1.00	1.00	1.00	0.00	0.00	1.00	0.25	1.00	1.00	1.00	1.00	1.00	0.04	0.12	0.04	0.65	1.00	1.00
0.46	1.00	1.00	0.00	0.00	0.00	0.00	0.25	0.25	1.00	0.00	1.00	1.00	0.02	0.09	0.04	0.62	0.00	1.00
0.81	0.00	0.00	1.00	0.00	1.00	1.00	0.25	0.25	1.00	0.00	1.00	1.00	0.12	0.39	0.05	0.33	0.13	1.00
0.95	1.00	1.00	1.00	0.00	1.00	1.00	1.00	0.25	1.00	1.00	1.00	1.00	0.10	0.08	0.01	0.65	1.00	1.00

Tabel 3.6 Data uji hasil perhitungan Kernel

Atr 1	Atr 2	Atr 3	Atr 4	Atr 5	Atr 6	Atr 7	Atr 8	Atr 9	Atr 10	Atr 11	Atr 12	Atr 13	Atr 14	Atr 15	Atr 16	Atr 17	Atr 18	Atr 19
1.00	1.00	1.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
0.76	1.00	1.00	1.00	0.00	0.00	0.00	1.00	0.25	1.00	0.00	1.00	1.00	0.40	0.34	0.85	0.91	0.00	0.00
0.52	1.00	0.00	1.00	0.00	1.00	1.00	0.25	0.25	0.00	1.00	0.00	0.00	0.37	1.00	0.20	0.33	0.14	0.00
0.81	1.00	1.00	1.00	0.00	0.00	1.00	0.25	1.00	1.00	0.00	0.00	1.00	1.00	0.28	0.11	0.62	0.12	0.00
0.21	1.00	1.00	1.00	0.00	1.00	1.00	0.25	0.25	1.00	1.00	1.00	1.00	0.02	0.16	1.00	0.91	0.55	1.00
0.17	1.00	0.00	1.00	1.00	1.00	1.00	0.25	1.00	1.00	1.00	1.00	1.00	0.02	0.12	0.06	0.95	1.00	0.00
0.17	1.00	1.00	0.00	1.00	1.00	1.00	1.00	0.25	1.00	1.00	1.00	1.00	0.02	0.20	0.03	1.00	0.53	1.00
0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.25	1.00	1.00	1.00	1.00	1.00	0.04	0.10	0.01	1.00	0.57	1.00

Setelah dilakukan perhitungan nilai kernel pada data uji dan latih, maka selanjutnya akan dilakukan perhitungan training SVM pada latih. Untuk proses training pada SVM yaitu dengan mencari nilai w (*weight*), nilai b (*bias*) dan nilai *slack variable*. Untuk mencari nilai parameter tersebut dilakukan proses *Quadratic Programming*, maka dapat dibuat model *Quadratic Programming* dengan mengimplementasikan persamaan 2.9 berikut :

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l t_i$$

$$\text{subject to } y_i(x_i \cdot w + b) + t_i \geq 1,$$

$$t_i \geq 0, i = 1, \dots, l$$

Maksud dari dari persamaan 2.9 adalah meminimumkan nilai fungsi *objective*

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^l t_i \text{ dengan batas (constraint) } y_i(x_i \cdot w + b) + t_i \geq 1.$$

Minimumkan fungsi *Objective*

$$\frac{1}{2}(w_1^2 + w_2^2 + w_3^2 + w_4^2 + w_5^2 + w_6^2 + w_7^2 + w_8^2 + w_9^2 + w_{10}^2 + w_{11}^2 + w_{12}^2 + w_{13}^2 + w_{14}^2 + w_{15}^2 + w_{16}^2 + w_{17}^2 + w_{18}^2 + w_{19}^2) + C * (t_1 + t_2 + t_3 + t_4 + t_5 + t_6 + t_7 + t_8 + t_9 + t_{10} + t_{11} + t_{12} + t_{13} + t_{14} + t_{15} + t_{16} + t_{17} + t_{18} + t_{19})$$

Dengan fungsi Batasan (*Constraint*):

$$-1(0.53 \cdot w_1 + 1 \cdot w_2 + 1 \cdot w_3 + 1 \cdot w_4 + 1 \cdot w_6 + 0.25 \cdot w_8 + 1 \cdot w_9 + 1 \cdot w_{12} + 1 \cdot w_{13} + 1 \cdot w_{19} + b) + t_1 \geq 1$$

$$-1(0.18 \cdot w_1 + 1 \cdot w_2 + 1 \cdot w_3 + 1 \cdot w_7 + 0.25 \cdot w_8 + 0.25 \cdot w_9 + 1 \cdot w_{10} + 1 \cdot w_{11} + 1 \cdot w_{12} + 1 \cdot w_{13} + 0.31 \cdot w_{14} + 1 \cdot w_{15} + 0.16 \cdot w_{16} + 0.65 \cdot w_{17} + 0.16 \cdot w_{18} + 1 \cdot w_{19} + b) + t_2 \geq 1$$

$$-1(1 \cdot w_1 + 1 \cdot w_2 + 1 \cdot w_3 + 1 \cdot w_7 + 1 \cdot w_{10} + 1 \cdot w_{11} + 1 \cdot w_{12} + 1 \cdot w_{13} + 0.06 \cdot w_{14} + 1 \cdot w_{15} + 0.06 \cdot w_{16} + 1 \cdot w_{19} + b) + t_3 \geq 1$$

$$-1(0.14 \cdot w_1 + 1 \cdot w_2 + 1 \cdot w_4 + 1 \cdot w_6 + 1 \cdot w_7 + 0.25 \cdot w_8 + 1 \cdot w_9 + 1 \cdot w_{10} + 1 \cdot w_{12} + 1 \cdot w_{13} + 0.04 \cdot w_{14} + 0.06 \cdot w_{15} + 0.01 \cdot w_{16} + 0.08 \cdot w_{17} + 1 \cdot w_{19} + b) + t_4 \geq 1$$

Dan seterusnya....

$$1(0.02 \cdot w_1 + 1 \cdot w_2 + 1 \cdot w_4 + 1 \cdot w_6 + 1 \cdot w_7 + 0.25 \cdot w_8 + 1 \cdot w_9 + 1 \cdot w_{10} + 1 \cdot w_{11} + 1 \cdot w_{12} + 1 \cdot w_{13} + 0.04 \cdot w_{14} + 0.12 \cdot w_{15} + 0.04 \cdot w_{16} + 0.65 \cdot w_{17} + 1 \cdot w_{18} + 1 \cdot w_{19} + b) + t_{17} \geq 1$$

$$1(0.46 \cdot w_1 + 1 \cdot w_2 + 1 \cdot w_3 + 0.25 \cdot w_8 + 0.25 \cdot w_9 + 1 \cdot w_{10} + 1 \cdot w_{12} + 1 \cdot w_{13} + 0.02 \cdot w_{14} + 0.09 \cdot w_{15} + 0.04 \cdot w_{16} + 0.62 \cdot w_{17} + 1 \cdot w_{19} + b) + t_{18} \geq 1$$

$$1(0.81 \cdot w_1 + 1 \cdot w_4 + 1 \cdot w_6 + 1 \cdot w_7 + 0.25 \cdot w_8 + 0.25 \cdot w_9 + 1 \cdot w_{10} + 1 \cdot w_{12} + 1 \cdot w_{13} + 0.12 \cdot w_{14} + 0.39 \cdot w_{15} + 0.05 \cdot w_{16} + 0.33 \cdot w_{17} + 0.13 \cdot w_{18} + 1 \cdot w_{19} + b) + t_{19} \geq 1$$

$$1(0.95 \cdot w_1 + 1 \cdot w_2 + 1 \cdot w_3 + 1 \cdot w_4 + 1 \cdot w_6 + 1 \cdot w_7 + 1 \cdot w_8 + 0.25 \cdot w_9 + 1 \cdot w_{10} + 1 \cdot w_{11} + 1 \cdot w_{12} + 1 \cdot w_{13} + 0.10 \cdot w_{14} + 0.08 \cdot w_{15} + 0.01 \cdot w_{16} + 0.65 \cdot w_{17} + 1 \cdot w_{18} + 1 \cdot w_{19} + b) + t_{20} \geq 1$$

Untuk menyelesaikan model *Quadratic Programming* di atas membutuhkan perhitungan yang panjang. Oleh karena itu untuk mendapatkan nilai parameter SVM tersebut akan dibantu dengan bantuan *solver Quadratic Programming*. Secara manual untuk menyelesaikan *quadratic programming* yaitu dengan. Untuk perhitungan manual pada penelitian ini menggunakan bantuan *Microsoft solver Foundation*. Berdasarkan bantuan dari solver dengan nilai $C = 20$, maka didapatkan nilai w (*weight*), b (*bias*), serta *slack variable* t_i seperti pada tabel 3.7

Tabel 3.7 Nilai w (*weight*) dan b (*bias*) yang diperoleh

w1	-3.91E-13
w2	-2.43E-13
w3	4.70E-14
w4	-2.11E-13
w5	1.40E-03
w6	9.97E-13
w7	1.72E-13
w8	-2.35E-12
w9	2.92E-13
w10	2.62E-13
w11	3.92E-13
w12	2.76E-13
w13	5.24E-13
w14	-1.17E-11
w15	-2.26E-12
w16	-5.75E-12
w17	-1.09E-13
w18	-5.42E-12
w19	2.85E-13
b	-0.000414

Untuk *slack variable* didapatkan seperti pada tabel 3.8

Tabel 3.8 *Slack Variable*

t1	0.9996
t2	0.9996
t3	0.9996
t4	0.9996
t5	0.9996
t6	0.9996
t7	0.9996
t8	0.9996
t9	0.9996
t10	0.9996
t11	1.0004
t12	1.0004
t13	1.0004
t14	1.0004
t15	1.0004
t16	1.0004
t17	1.0004
t18	1.0004
t19	1.0004
t20	1.0004

Dari nilai w (*weight*) dan nilai *bias* dapat dihitung nilai fungsi *objective* sebagai berikut :

$$\begin{aligned} & \frac{1}{2}(w1^2 + w2^2 + w3^2 + w4^2 + w5^2 + w6^2 + w7^2 + w8^2 + w9^2 + w10^2 \\ & + w11^2 + w12^2 + w13^2 + w14^2 + w15^2 + w16^2 + w17^2 \\ & + w18^2 + w19^2) + C * (t1 + t2 + t3 + t4 + t5 + t6 + t7 + t8 \\ & + t9 + t10 + t11 + t12 + t13 + t14 + t15 + t16 + t17 + t18 \\ & + t19 + t20) \end{aligned}$$

$$\begin{aligned} & \frac{1}{2}((-3.91(E - 13))^2 + (-2.43(E - 13))^2 + (4.70(E - 14))^2 \\ & + (-2.11(E - 13))^2 + (1.40(E - 03))^2 + (9.97(E - 13))^2 \\ & + (1.72(E - 13))^2 + (-2.35(E - 12))^2 + (2.92(E - 13))^2 \\ & + (2.62(E - 13))^2 + (3.92(E - 13))^2 + (2.76(E - 13))^2 \\ & + (5.24(E - 13))^2 + (-1.17(E - 11))^2 + (-2.26(E - 12))^2 \\ & + (-5.75(E - 12))^2 + (-1.09(E - 13))^2 + (-5.42(E - 12))^2 \\ & + (2.85(E - 13))^2) + 20 \\ & * (0.9996 + 0.9996 + 0.9996 + 0.9996 + 0.9996 + 0.9996 \\ & + 0.9996 + 0.9996 + 0.9996 + 0.9996 + 1.0004 + 1.0004 \\ & + 1.0004 + 1.0004 + 1.0004 + 1.0004 + 1.0004 + 1.0004 \\ & + 1.0004 + 1.0004) = 400.00232 \end{aligned}$$

Dari tabel 3.7 dapat dilihat bahwa nilai w (*weight*) yang diperoleh sangat kecil, maka kemudian nilai w (*weight*) akan dinormalisasi dengan mengalikan dengan pangkat Eksponen yang dipilih, hal ini bertujuan untuk membuat proses perhitungan nilai *constraint* tidak menjadi sangat kecil, sehingga tetap berada pada rentang data yang sudah ada. Pada proses ini nilai *weight* dikalikan dengan nilai Eksponen = 20 .Adapun nilai w (*weight*) yang diperoleh dapat dilihat pada tabel 3.9

Tabel 3.9 Nilai weight hasil normalisasi

w1	-0.00018968
w2	-0.00011769
w3	0.00002282
w4	-0.00010256
w5	680510.40271618
w6	0.00048374
w7	0.00008332
w8	-0.00113861
w9	0.00014190
w10	0.00012717
w11	0.00019015
w12	0.00013402
w13	0.00025435
w14	-0.00567382
w15	-0.00109521
w16	-0.00278826
w17	-0.00005280
w18	-0.00263102
w19	0.00013843

Selanjutnya adalah menghitung nilai *score* untuk menentukan nilai klasifikasi dari record yang ada dengan mengimplementasikan persamaan 2.4

$$g(x) = \text{sign} (f(x))$$

$$\text{dengan } f(x) = (w^T x + b),$$

sebelumnya hitung nilai *constraint* dengan pada setiap record data, dimana *constraint* adalah

$$f(x) = y_i(x_i \cdot w + b)$$

$$\begin{aligned} & -1((0.526 * (-0.00018968)) + (1 * (-0.00011769)) + (1 * 0.00002282) \\ & + (1 * (-0.00010256)) + (0 * 680510.402) + (1 \\ & * (0.00048374)) + (0 * 0.00008332) + (0.25 \\ & * (-0.00113861)) + (1 * 0.00014190) + (0 * 0.00012717) \\ & + (0 * (-0.00019015)) + (1 * (0.00013402)) + (1 \\ & * (-0.00567382)) + (0 * (-0.00109521)) + (0 \\ & * (-0.00278826)) + (0 * (-0.00005280)) + (0 \\ & * (-0.00263102)) + (1 * (0.00013843)) + (-0.000414) + 0)) \\ & = -0.000156259 \end{aligned}$$

Selanjutnya nilai *score* dapat dihitung sebagai berikut :

$$g(x) = \text{sign} (w^T x + b)$$

$$\begin{aligned} & \text{Sign}(-1((0.526 * (-0.00018968)) + (1 * (-0.00011769)) + (1 \\ & * 0.00002282) + (1 * (-0.00010256)) + (0 * 680510.402) + (1 \\ & * (0.00048374)) + (0 * 0.00008332) + (0.25 \\ & * (-0.00113861)) + (1 * 0.00014190) + (0 * 0.00012717) \\ & + (0 * (-0.00019015)) + (1 * (0.00013402)) + (1 \\ & * (-0.00567382)) + (0 * (-0.00109521)) + (0 \\ & * (-0.00278826)) + (0 * (-0.00005280)) + (0 \\ & * (-0.00263102)) + (1 * (0.00013843)) + (-0.000414) + 0)) \\ & = -1 \end{aligned}$$

Adapun hasil perhitungan nilai *constraint* dan nilai *score* klasifikasi pada data latih dapat dilihat pada tabel 3.10

Tabel 3.10 Data nilai *constraint* dan *score* pada data latih

Constraint	score
-0.000156259	0.000156259
0.00365872	-0.00365872
0.000382992	-0.000382992
-0.000153247	0.000153247
0.007604417	-0.007604417
0.005443315	-0.005443315
0.005351554	-0.005351554
0.001518755	-0.001518755
0.003467855	-0.003467855
0.006093424	-0.006093424
680510.4012	680510.4012
-0.000590426	-0.000590426
-0.0028447	-0.0028447
680510.4002	680510.4002
680510.4022	680510.4022
680510.4022	680510.4022
-0.002938474	-0.002938474
-0.000555754	-0.000555754
-0.001290541	-0.001290541
-0.003834799	-0.003834799

Tahap 3 : Proses Pengujian dengan SVM

Setelah data dibuat model dari proses training, maka kemudian akan dilakukan pengujian dengan menggunakan data uji. Data uji sebelumnya telah dilakukan perhitungan kernel, adapun hasil perhitungan kernel tersebut dapat dilihat pada tabel 3.6. Data uji yang telah dilakukan proses kernel akan dilakukan tahap pengujian SVM sesuai model yang telah didapatkan.

Kemudian nilai atribut pada setiap record akan dihitung nilai *score* untuk mendapatkan nilai klasifikasi, dengan mengimplementasikan persamaan (2.4).

$$g(x) = \text{sign} (f(x))$$

dengan $f(x) = (w^T x + b)$,

fungsi sign untuk record ke -1 pada data latih :

$$f(x) = (w^T x + b)$$

$$\begin{aligned}
& ((1.(-0.00018968)) + (1.(-0.00011769)) + (1.(0.00002282))) \\
& + (1.(-0.00010256)) + (0.(680510.402)) \\
& + (1.(0.00048374)) + (0.(0.00008332)) \\
& + (0.25.(-0.00113861)) + 1.(0.00014190) \\
& + 0.(0.00012717) + 0.(0.00019015) + 1.(0.00013402) \\
& + 1.(0.00025435) + 0.(-0.00567382) + 0.(-0.00109521) \\
& + 0.(-0.00278826) + 0.(-0.00005280) \\
& + 0.(-0.00263102) + 1.(0.00013843) + (-0.000414) \\
& = -0.000202647
\end{aligned}$$

$$\text{sign}(-0.000202647) = -1$$

Adapun hasil perhitungan *score* dan klasifikasi dapat dilihat pada tabel 3.11

Tabel 3.11 Data Hasil perhitungan *score* dan nilai klasifikasi pada data uji

Score	Classification
-0.000202647	-1
-0.006384418	-1
-0.004381107	-1
-0.007089269	-1
-0.004043476	-1
680510.4001	1
680510.4006	1
-0.001053046	-1

Tabel 3.12 Data Uji dengan nilai klasifikasi

Atr 2	Atr 3	Atr 4	Atr 5	Atr 6	Atr 7	Atr 8	Atr 9	Atr 10	Atr 11	Atr 12	Atr 13	Atr 14	Atr 15	Atr 16	Atr 17	Atr 18	Atr 19	Class
1.00	1.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	-1
1.00	1.00	1.00	0.00	0.00	0.00	1.00	0.25	1.00	0.00	1.00	1.00	0.40	0.34	0.85	0.91	0.00	0.00	-1
1.00	0.00	1.00	0.00	1.00	1.00	0.25	0.25	0.00	1.00	0.00	0.00	0.37	1.00	0.20	0.33	0.14	0.00	-1
1.00	1.00	1.00	0.00	0.00	1.00	0.25	1.00	1.00	0.00	0.00	1.00	1.00	0.28	0.11	0.62	0.12	0.00	-1
1.00	1.00	1.00	0.00	1.00	1.00	0.25	0.25	1.00	1.00	1.00	1.00	0.02	0.16	1.00	0.91	0.55	1.00	-1
1.00	0.00	1.00	1.00	1.00	1.00	0.25	1.00	1.00	1.00	1.00	1.00	0.02	0.12	0.06	0.95	1.00	0.00	1
1.00	1.00	0.00	1.00	1.00	1.00	1.00	0.25	1.00	1.00	1.00	1.00	0.02	0.20	0.03	1.00	0.53	1.00	1
0.00	0.00	0.00	0.00	1.00	1.00	0.25	1.00	1.00	1.00	1.00	1.00	0.04	0.10	0.01	1.00	0.57	1.00	-1

Dari hasil klasifikasi di atas dapat diambil keputusan bahwa data dengan nilai klasifikasi -1 masuk dalam kategori hepatitis kronis (*die*), sedangkan data dengan nilai klasifikasi 1 maka akan masuk dalam kategori hepatitis akut (*live*).

Untuk evaluasi dapat dilakukan dengan membuat *confusion table* seperti pada tabel 3.13.

Tabel 3.13 *Confusion Table*

True class	Prediction class		
	1	-1	Jumlah
1	2	0	2
-1	2	4	6

Untuk menghitung akurasi digunakan persamaan (2.24)

$$\begin{aligned}
 \text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \\
 &= \frac{2 + 4}{2 + 4 + 0 + 2} \times 100\% \\
 &= 75\%
 \end{aligned}$$

$$\text{Sensitivity} = \frac{TP}{(TP + FN)} = \frac{2}{2 + 2} = 0.5$$

$$\text{Specifity} = \frac{TN}{(TN + FP)} = \frac{4}{4 + 0} = 1$$

3.6 Rancangan Uji Coba

Setelah sistem dibuat, kemudian maka langkah selanjutnya adalah melakukan pengujian terhadap sistem tersebut. Pengujian dilakukan untuk mengetahui tingkat akurasi dari hasil klasifikasi penyakit hepatitis dengan menggunakan metode SVM. Pada dataset *Hepatitis Domain* sudah tersedia hasil pengelompokkan masing – masing *record*. Hasil kelas ini akan digunakan sebagai

perbandingan dengan hasil oleh sistem. Pengujian dilakukan terhadap jumlah data latih terhadap tingkat akurasi klasifikasi.

3.6.1 Uji Pengaruh Jumlah Data latih terhadap akurasi proses klasifikasi

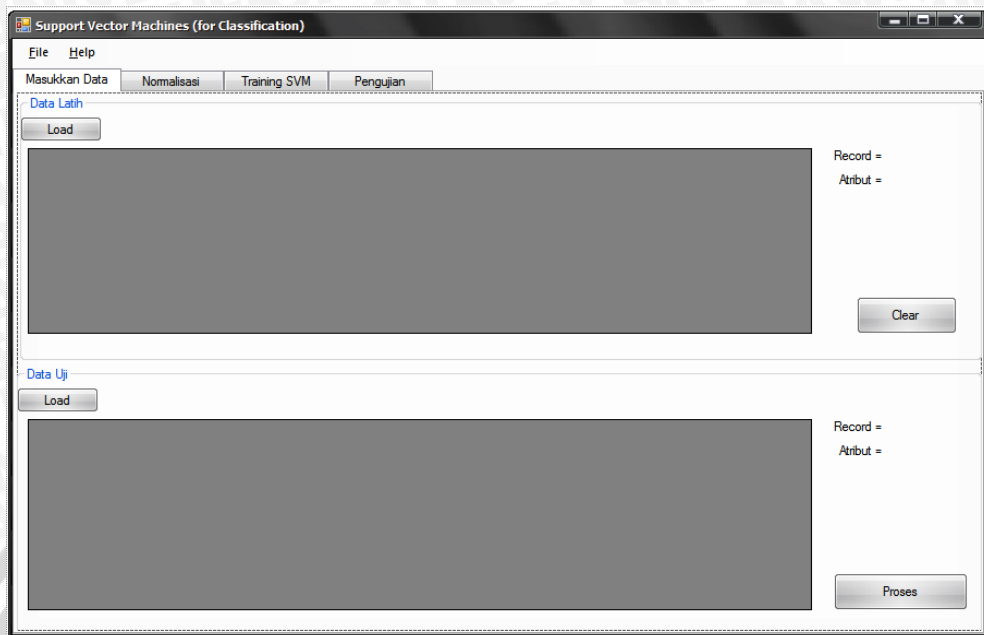
Uji pengaruh data latih terhadap tingkat akurasi proses klasifikasi dilakukan dengan mengambil sejumlah data latih dan data uji. Data latih yang digunakan adalah 20, 40, 60, 80 Data uji yang digunakan pada penelitian ini adalah sebanyak 75 yang dipilih secara random dari dataset. Perhitungan tingkat akurasi akan menggunakan persamaan 2.24. Rancangan tabel yang akan digunakan untuk mencatat hasil pengujian akan ditampilkan pada tabel 3.12.

Tabel 3.12 Uji Hasil Klasifikasi

No	Jumlah data Latih	C	Akurasi (%)	Specificity	Sensitivity
1	20	20	75%	50%	100%
· · dst.					

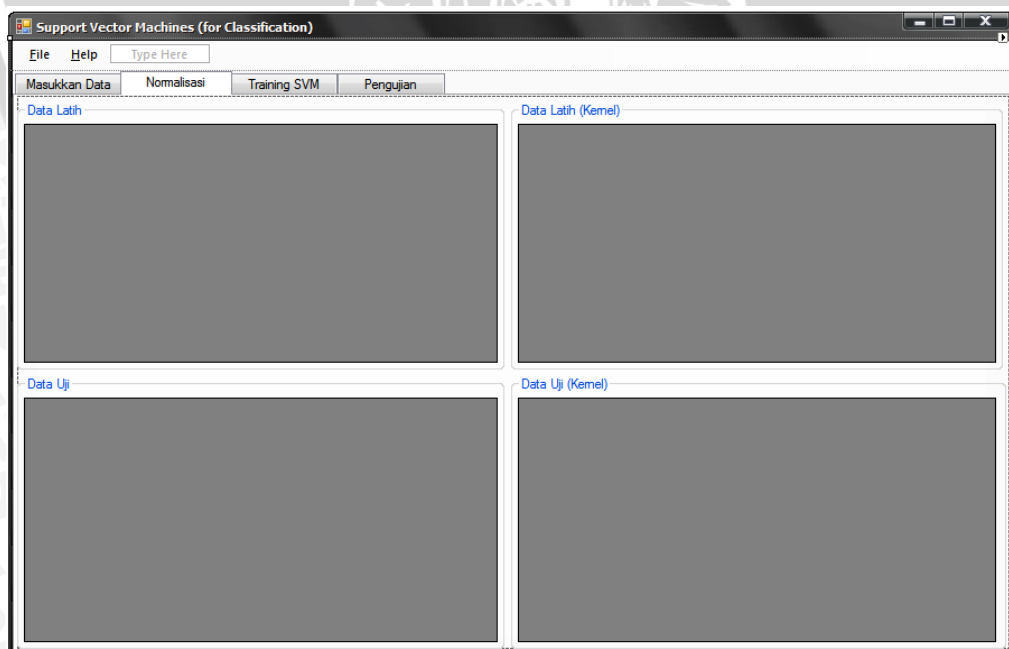
3.6.2 Perancangan Antarmuka

Antarmuka yang dirancang pada sistem ini yaitu terdiri dari 4 komponen, bagian – bagian tersebut adalah, bagian *input* data, merupakan bagian untuk input data latih dan data uji, bagian Normalisasi, merupakan bagian untuk melakukan normalisasi terhadap data uji dan data latih, dan kemudian dilakukan proses kernel secara linear, bagian training SVM adalah bagian untuk mencari nilai – nilai parameter SVM , dan bagian pengujian adalah proses pengujian dan pengukuran tingkat akurasi dari metode SVM terhadap data uji dan data latih. Adapun tampilan antarmuka dari sistem ditunjukkan pada gambar 3.8 sampai gambar 3.11.



Gambar 3.8 Form *input* data

Pada gambar 3.8 merupakan tampilan dari form untuk proses input data, baik data uji maupun data latih. Terdapat tombol *Load* untuk memasukkan data ke dalam form untuk diproses. Tombol *clear* berfungsi untuk mengosongkan *gridview* pada data latih maupun pada data uji. Sedangkan tombol proses untuk melakukan proses Normalisasi pada data uji maupun data latih.

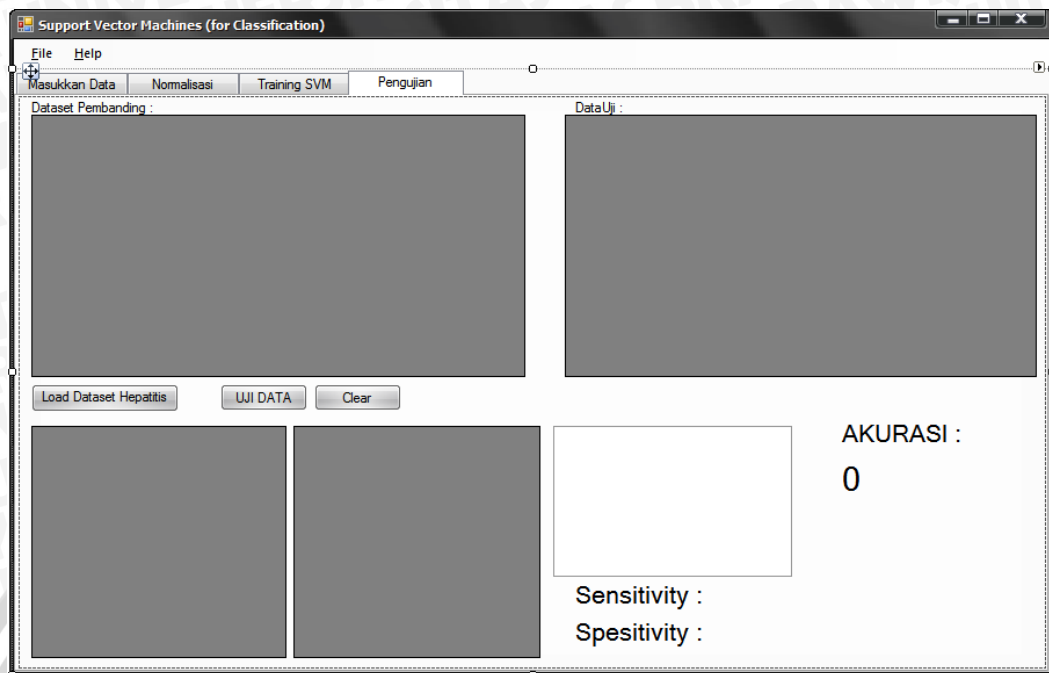


Gambar 3.9 Form Normalisasi dan Kernel Data

Pada gambar 3.9 merupakan form untuk menampilkan data latih dan data uji yang telah dilakukan normalisasi, dan juga hasil dari kernel dari data uji dan data latih yang dilakukan secara linear. Hasil dari proses Normalisasi pada data latih dan data uji akan ditampilkan pada *gridview* data latih dan data uji. Sedangkan untuk hasil Kernel akan ditampilkan pada *gridview* kernel data latih maupun data uji.

Gambar 3.10 Form Training SVM

Pada form ini merupakan tampilan untuk melakukan proses training SVM, dimana pada proses training akan didapatkan nilai – nilai parameter SVM sesuai dengan data latih yang diinputkan. Bagian – bagian dari form dapat dilihat pada gambar 3.10, dimana terdapat bagian fungsi *objective* untuk melihat nilai dari fungsi *objective* yang dihasilkan, kemudian nilai *b* (bias). Nilai *w* (*weight*) dari atribut yang dihasilkan dapat dilihat pada *textbox weight*. Pada *gridview* model akan menampilkan nilai *constraint* serta *score* klasifikasi dari data latih. Tombol optimasi berfungsi untuk melakukan proses training SVM untuk mendapatkan nilai parameter SVM, sedangkan tombol *clear* untuk mengosongkan *textbox* dan *gridview*.



Gambar 3.11 Form Pengujuan

Pada gambar 3.11 merupakan tampilan form pengujuan untuk data uji, serta untuk mengukur tingkat akurasi dari model yang telah dibuat. Untuk proses pengujuan yaitu dengan membandingkan nilai data uji yang berisi kelas asli dengan data uji hasil proses dengan SVM. Tombol *Load* berfungsi untuk memasukkan data uji yang berisi kelas asli ke dalam *gridview*, sedangkan pada *gridview* data uji berisi data uji dengan kelas hasil proses SVM. Pada textbox akan menampilkan *confusion table* berupa nilai *True Positive*, *True Negative*, *False Positive* dan *False Negatif*. Kemudian akan ditampilkan juga tingkat akurasi, nilai sensitivity dan spesitivity.

