#### **BAB 4 PERANCANGAN**

Pada bab ini berisi mengenai alur penyelesaian masalah pengklasifikasian jenis penyakit Skizofrenia. Bab ini terdiri dari bagian yaitu formulasi permasalahan, alur klasifikasi metode *Support Vector Machine* (SVM), perhitungan manual, dan perancangan *user interface*.

# 4.1 Deskripsi Sistem

Sistem yang akan dibuat adalah mengenai klasifikasi jenis penyakit Skizofrenia yang terbagi menjadi lima kelas. Kelas 1 adalah jenis *Paranoid* (F20.0), kelas 2 adalah jenis *Hebefrenik* (F20.1), kelas 3 adalah jenis Katatonik (F20.2), kelas 4 adalah jenis *Undifferentiated* (F20.3), dan kelas 5 adalah jenis Simpleks (F20.6). Data yang digunakan dalam proses klasifikasi adalah data rekam medis dari pasien penyakit Skizofrenia. Data ini akan digunakan sebagai data latih dan data uji. Parameter yang digunakan dalam perhitungan berasal dari gejala-gejala yang ada dalam diagnosis penyakit Skizofrenia.

Setelah itu proses klasifikasi akan menggunakan metode *Support Vector Machine* (SVM). Pada klasifikasi ini menggunakan *kernel polynomial*, karena data yang digunakan bersifat nonlinier. Selajutnya data latih akan diproses dengan menggunakan *Sequential Training SVM*. Dalam menyelesaikan permasalahan yang bersifat multikelas, maka konsep *One Against All* akan digunakan untuk menentukan kelas pada data latih. Hasil yang akan didapatkan dalam proses *training* adalah nilai  $\alpha_i$  dan nilai bias dari setiap level.

Pada proses manualisasi digunakan data sebanyak 7 data. Data ini terdiri dari 5 data latih dan 2 data uji. Setiap satu data latih mewakili tiap kelas pada data. Data yang digunakan terdapat pada Tabel 4.1

Fungsi Kognitif Perilaku Terhadap Pemeriksa Atensi dan Memori No Kelas Konsentrasi G1 G2 G3 G4 G5 G85 G86 G87 G88 ... 

**Tabel 4.1 Data Rekam Medis Skizofrenia** 

Parameter G1 hingga G88 merupakan atribut berdasarkan gejala yang ada. Nilai pada data tersebut bernilai biner, karena gejala hanya bernilai 1 untuk

"YA" dan o untuk "TIDAK". Data ini didapatkan dari RSJ. Radjiman Wediodiningrat, dimana dalam menentukan gejala sebagai parameter dilakukan konsultasi dengan pakar yaitu seorang dokter spesialis kejiwaan.

#### 4.2 Transformasi Data

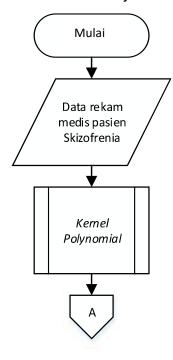
Proses transformasi data merupakan proses yang dibutuhkan dalam melakukan data *mining*. Proses ini bertujuan untuk mengubah data yang akan digunakan pada tahap selanjutnya yaitu perhitungan klasifikasi nanti. Transfomasi data pada penelitian ini adalah berupa pemberian nilai pada setiap data, dimana untuk nilai bobot pada setiap data hanya akan bernilai 1 atau 0. Nilai 1 menyatakan bahwa pasien mengalami gejala tersebut, sedangkan 0 menyatakan bahwa pasien tidak mengalami gejala tersebut. Pada tahap ini data akan dibagi dalam 88 parameter. Setiap parameter merepresentasikan gejala yang digunakan dalam mendiagnosis penyakit skizofrenia. Daftar kode gejala yang akan dijadikan sebagai parameter ditampilkan pada Lampiran A.

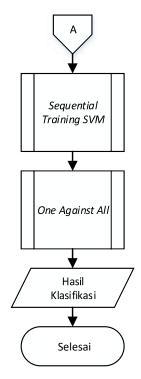
### 4.3 Alur Support Machine Vector

Pada bagian ini akan dijelaskan alur dari proses metode *Support Vector Machine*, mulai dari proses perhitungan *kernel* hingga pengujian.

### 4.3.1 Proses Support Vector Machine

Pada langkah awal dalam proses SVM akan dilakukan *input* data dari *database*. Setelah itu data akan dihitung menggunakan *kernel polynomial* untuk diteruskan ke proses *training*. *Output* yang akan dihasilkan adalah berupa hasil klasifikasi. Gambar 4.1 akan menjelaskan proses ini.



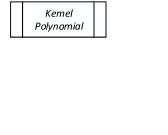


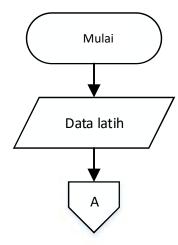
**Gambar 4.1 Proses SVM** 

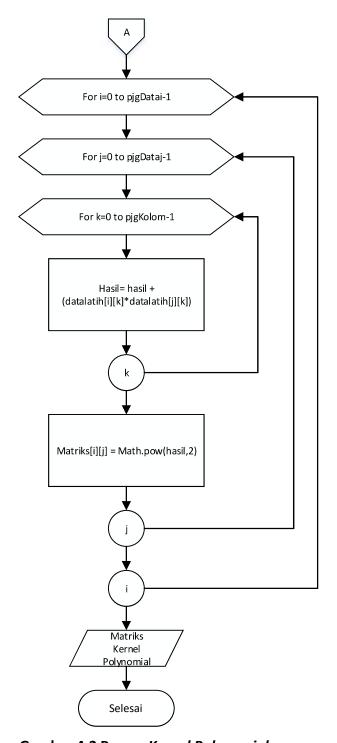
- 1. Input yang digunakan adalah data rekam medis pasien Skizofrenia
- 2. Proses perhitungan data latih ke bentuk kernel
- 3. Proses training data dengan metode sequential training SVM
- 4. Proses konsep One Against All untuk menentukan level pada setiap kelas
- 5. Output dari proses ini adalah hasil prediksi dari kelas pada data yang diuji

# 4.3.2 Kernel Polynomial

Tahap ini merupakan tahap dimana akan dilakukan proses perhitungan kernel polynomial. Tahapan pada proses perhitungan kernel polynomial akan digambarkan pada Gambar 4.2





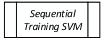


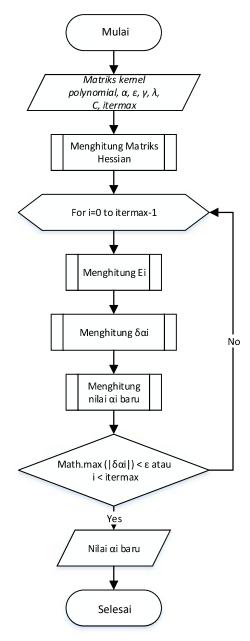
**Gambar 4.2 Proses Kernel Polynomial** 

- 1. Input yang digunakan adalah data latih.
- 2. Proses perkalian antar data latih secara berulang sebanyak jumlah data latih
- 3. Hasil dari perkalian lalu dipangkatkan 2
- 4. Output dari proses ini adalah nilai dari perhitungan kernel polynomial

# 4.3.3 Sequential Training SVM

Proses *sequential training* SVM adalah proses yang digunakan untuk mencari nilai *hyperplane* terbaik. Proses *Sequential training* SVM disajikan dalam Gambar 4.3.





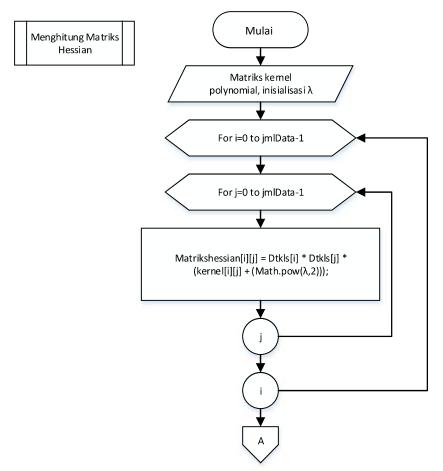
Gambar 4.3 Sequential Training SVM

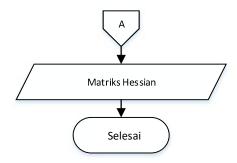
1. Input yang digunakan adalah nilai matriks kernel polynomial dan inisialisasi nilai  $\alpha_i$  (fungsi lagrange),  $\gamma$  (gamma),  $\varepsilon$  (epsilon), C (Complexity),  $\lambda$  (lambda), dan itermax.

- 2. Setelah input data dan inisialisasi, akan dilakukan perhitungan matriks *hessian*
- 3. Langkah selanjutnya adalah melakukan perhitungan untuk mencari nilai  $E_i$  (nilai *error*) dengan Persamaan 2.11.
- 4. Lalu melakukan perhitungan unutk mencari nilai  $\delta \alpha_i$  dengan Persamaan 2.12.
- 5. Selanjutnya melakukan perhitungan untuk mencari nilai  $\alpha_i$  yang baru dengan Persamaan 2.13
- 6. Langkah 3 hingga langkah 5 akan terus diulang hingga mencapai nilai konvergen yang maksimal, yaitu ( $|\delta\alpha_i|$ ) <  $\varepsilon$  atau hingga mencapai nilai itermax.
- 7. Output dari proses ini adalah nilai  $\alpha_i$  yang merupakan nilai hyperplane terbaik. Nilai hyperplane ini akan menjadi Support Vector, yang dimana akan digunakan ketika proses pengujian nanti.

#### 4.3.4 Matriks Hessian

Tahapan proses untuk mendapatkan nilai pada Matriks *Hessian* terdapat pada Gambar 4.4



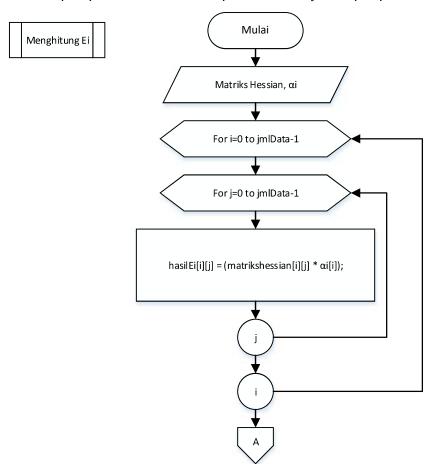


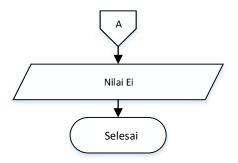
Gambar 4.4 Matriks Hessian

- 1. Input yang digunakan adalah data pada Matriks kernel polynomial dan nilai λ.
- Proses perhitungan untuk mendapatkan nilai pada matriks hessian dengan Persamaan 2.10.
- 3. Proses perulangan sebanyak jumlah data pada matriks kernel polynomial.
- 4. Ouput dari proses ini adalah nilai matriks Hessian.

# 4.3.5 Perhitungan Nilai $\boldsymbol{E_i}$

Tahapan proses untuk mendapatkan nilai  $E_i$  terdapat pada Gambar 4.5.



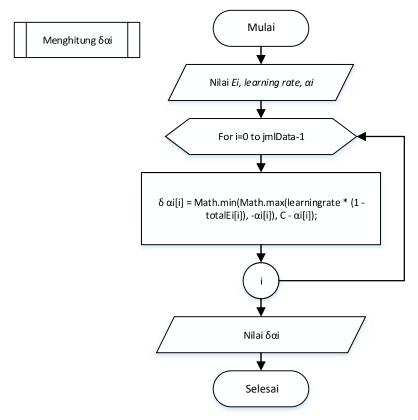


Gambar 4.5 Proses perhitungan nilai  $E_i$ 

- 1. Input yang digunakan adalah matriks hessian dan nilai  $\alpha_i$ .
- 2. Proses perhitungan untuk mendapatkan nilai  $E_i$  dengan Persamaan 2.11.
- 3. Proses perulangan sebanyak jumlah pada data matriks Hessian.
- 4. Output dari proses ini adalah nilai  $E_i$ .

# 4.3.6 Perhitungan Nilai $\delta_{lpha i}$

Tahapan proses untuk mendapatkan nilai  $\delta \alpha_i$  terdapat pada Gambar 4.6.



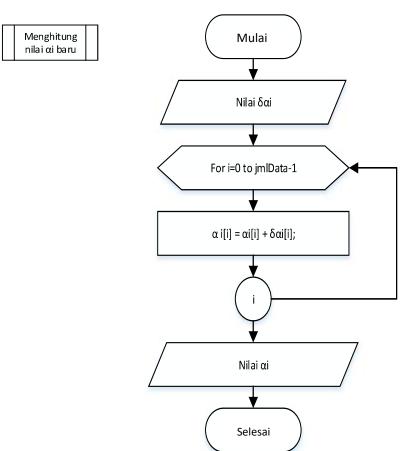
Gambar 4.6 Proses Perhitungan nilai  $\delta lpha_i$ 

- 1. Input yang digunakan adalah nilai  $E_i$ , learning rate, dan nilai  $\alpha_i$ .
- 2. Proses perhitungan untuk mendapatkan nilai  $\delta \alpha_i$  dengan Persamaan 2.12.

- 3. Proses perulangan sebanyak jumlah data yang ada pada nilai  $E_i$ .
- 4. *Output* dari proses ini adalah nilai  $\delta \alpha_i$ .

# 4.3.7 Perhitungan Nilai $lpha_i$

Tahapan proses untuk mendapatkan nilai  $\alpha_i$  yang baru terdapat pada Gambar 4.7.

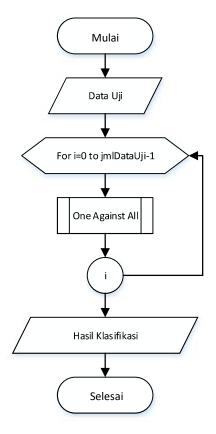


Gambar 4.7 Proses Perhitungan nilai  $lpha_i$ 

- 1. *Input* yang digunakan adalah nilai  $\delta_{\alpha i}$ .
- 2. Proses perhitungan untuk mendapatkan nilai  $\alpha_i$  dengan Persamaan 2.13.
- 3. Proses perulangan sebanyak jumlah data yang ada pada nilai  $\delta \alpha_i$ .
- 4. Output dari proses ini adalah nilai  $\alpha_i$  yang baru.

### 4.3.8 Proses Pengujian SVM

Proses pengujian merupakan tahapan dalam menguji klasifikasi pada data latih yang telah diproses sebelumnya. Dalam proses ini akan jenis pengujian *K-fold Cross Validation*, dimana akan digunakan 11 *fold* dalam pengujian ini. Tahapan dalam proses pengujian ditampilkan pada Gambar 4.8.



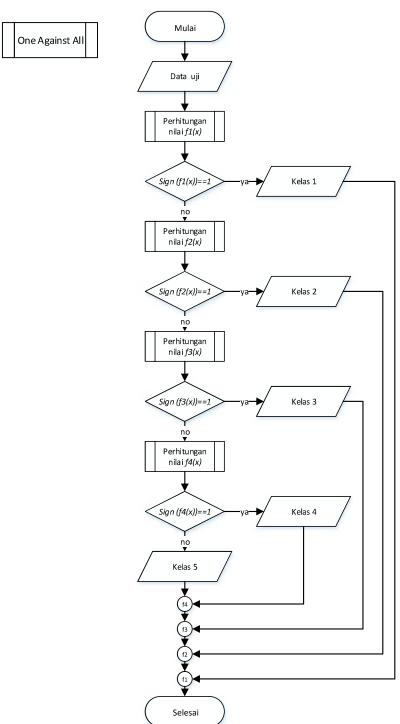
Gambar 4.8 Pengujian SVM

- 1. Input yang digunakan adalah data uji.
- 2. Untuk menyelesaikan permasalahan multikelas maka akan digunakan konsep *One Against All*.
- 3. Jika nilai uji tidak bernilai 1 atau bernilai -1 maka akan masuk ke level selanjutnya.
- 4. Output dari proses pengujian ini adalah hasil klasifikasi.

#### 4.3.9 One Against All

Konsep One Against All adalah konsep yang digunakan pada metode SVM dalam mengatasi permasalahan multikelas. Penelitian mengenai klasifikasi jenis penyakit Skizofrenia membutuhkan konsep One Against All, karena jenis penyakit Skizofrenia terbagi menjadi 5 kelas. Jenis-jenis penyakit Skizofrenia yaitu paranoid, hebefrenik, katatonik, undifferentiated, dan simpleks. Lalu untuk membedakan setiap kelas maka metode ini membutuhkan 4 level yang berbeda. Pada level 1 berfungsi untuk memisahkan kelas 1 (Paranoid) yang bernilai 1, dengan kelas lainnya yang bernilai -1. Setelah itu pada level 2 berfungsi untuk memisahkan kelas 2 (Hebrefenik) yang bernilai 1, dengan kelas lainnya yang bernilai -1. Lalu pada level 3 berfungsi untuk memisahkan kelas 3 (Katatonik) yang bernilai 1, dengan kelas lainnya yang bernilai -1. Pada level 4 berfungsi untuk memisahkan kelas (Undifferentiated) yang bernilai 1, dengan kelas 5 (Simpleks) yang bernilai -1. Selanjutnya untuk menentukan data uji termasuk

kelas yang mana, maka data uji akan diproses. Jika nilai data uji bernilai 1 pada level 1 maka data uji tersebut merupakan kelas 1, dan bila bernilai -1 maka dia akan melanjutkan proses perhitungan kembali ke level selanjutnya. Proses ini akan terus diulang hingga data uji bernilai 1 pada level tertentu, atau diproses hingga level terakhir yaitu level 4. Untuk nilai -1 pada level 4 maka otomatis masuk ke kelas 5. Gambar 4.9 menggambarkan proses konsep *One Against All.* 

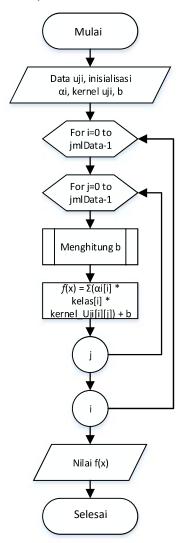


Gambar 4.9 Konsep One Against All

# 4.3.10 Perhitungan Nilai f(x)

Nilai f(x) berfungsi untuk menentukan kelas pada data uji. Proses dalam menghitung nilai f(x) ditampilkan pada Gambar 4.10.



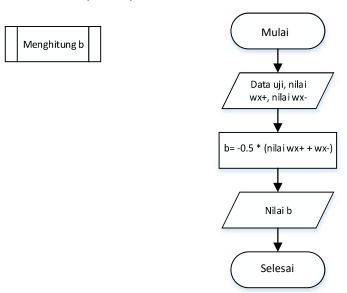


Gambar 4.10 Proses Perhitungan Nilai f(x)

- 1. Input yang digunakan adalah data uji, nilai  $\alpha_i$ , kernel data uji, nilai b (bias).
- 2. Proses perhitungan nilai b yang berfungsi sebagai nilai bias yang akan dibutuhkan dalam mengitung nilai f(x).
- 3. Setelah mendapat nilai b lalu masuk proses perhitungan nilai f(x) menggunakan persamaan  $f(x) = \alpha_i y_i K(x_i x_i)$ . Nilai  $y_i$  adalah nilai dari kelas setiap data uji.
- 4. Output dari proses ini adalah nilai f(x) setiap data uji.

### 4.3.11 Perhitungan Nilai b

Nilai b dibutuhkan dalam proses perhitungan nilai f(x). Proses perhitungan nilai b ditampilkan pada Gambar 4.11.



Gambar 4.11 Proses Perhitungan Nilai b

- 1. *Input* yang digunakan adalah data uji dan nilai  $wx^+$  (bobot kelas positif) dan nilai  $wx^-$  (bobot nilai negatif).
- 2. Proses perhitungan nilai b dengan Persamaan 2.3.
- 3. Outpur dar proses ini adalah nilai b

### 4.4 Perhitungan Manual Support Vector Machine

Proses perhitungan manual ini berfungsi dalam memberikan pemahaman dalam peyelesaian klasifikasi dengan metode SVM. Perhitungan manual ini akan diimplementasikan menjadi algoritma yang dipakai unutk membuat program klasifikasi ini. Dalam perhitungan manual ini data yang digunakan berasal dari data rekam medis pada RSJ Radjiman Wediodiningrat. Jumlah data yang digunakan pada proses perhitungan manual ini sebanyak 7 data, 5 data untuk data latih dan 2 untuk data uji. Kelas dibagi menjadi 5 kelas yang terdiri dari, kelas 1 untuk paranoid (F20.0), kelas 2 untuk hebrefenik (F20.1), kelas 3 untuk katatonik (F20.2), kelas 4 untuk undifferentiated (F20.3), dan kelas 5 untuk simpleks (F20.6). Parameter yang digunakan berasal dari gejala yang dipakai untuk mendiagnosa penyakit Skizofrenia, yang berjumlah sebanyak 88 parameter. Nilai pada setiap parameter akan bernilai biner, nilai 1 untuk mengalami dan nilai 0 untuk tidak mengalami.

#### 4.4.1 Proses Training SVM

Pada tahap awal dalam melakukan proses perhitungan manual pada SVM, proses training dibutuhkan untuk menghitung nilai hyperplane dan nilai margin terbaik. Hasil nilai pada proses training ini akan digunakan untuk melakukan

proses pengujian. Konsep yang digunakan dalam melakukan training adalah konsep sequential training SVM.

Tabel 4.2 adalah data latih yang akan digunakan dalam proses training. Jumlah data latih yang digunakan sebanyak 5 data yang merepresentasikan setiap kelas pada jenis penyakit. Data latih ke-1 merepresentasikan kelas paranoid, data latih ke-2 merepresentasikan kelas hebefrenik, data latih ke-3 merepresentasikan kelas katatonik, data latih ke-4 merepresentasikan kelas undifferentiated, data latih ke-5 merepresentasikan kelas simpleks. Setelah itu variabel G1 sampai G88 merupakan parameter yang digunakan. Pada data latih terdapat 5 kelas, maka dari itu dibutuhkan konsep One Against All. Konsep ini akan menerapkan perhitungan data hingga 5 level, untuk setiap kelas akan diberi nilai 1 untuk setiap level, sehingga kelas yang bukan pada level tersebut akan bernilai -1 lalu akan diteruskan perhitungan selanjutnya pada level berikutnya.

Data pada Tabel 4.2 juga akan dipakai untuk menghitung nilai pada kernel yang digunakan. Kernel yang digunakan pada perhitungan manual ini adalah kernel polynomial.

Fungsi Kognitif Perilaku terhadapa Pemeriksa Atensi dan NO Level Kelas Konsentrasi Memori G4 G87 G88 G1 G2 G3 G5 G85 G86 ... 1 1 0 0 0 1 2 2 1 0 0 0 0 1 0 1 0 -1 3 3 1 0 0 0 0 0 1 0 1 -1 4 4 0 -1 1 0 0 0 1 0 0 0 5 5 -1

Tabel 4.2 Data Latih Level 1

Tahap pertama dalam proses *training* adalah menghitung nilai *kernel polynomial*, hasil dari perhitungan nilai *kernel* ditampilkan pada Tabel 4.6.

Berikut ini merupakan contoh perhitungan *kernel polynomial* pertama dengan data latih ke-1 yang ditunjukan pada Tabel 4.3.

Tabel 4.3 Data Latih Ke-1

Data ke-	G1	G2	G3	G4	G5	 G85	G86	G87	G88
1	1	0	0	0	0	 1	0	1	0

$$k(x,y), \, {\rm misal} \, x = x_1, y = x_1 \, , {\rm maka} \, k(x,y) \colon$$
 
$$k(x_1.x_1) = (x_1.x_1)^2 = (13)^2 = 169$$
 
$$(x_1.x_1) = (1.1) + (0.0) + (0.0) + (0.0) + (0.0) + \cdots + (1.1) + (0.0) + (1.1) + (0.0)$$

$$= 1 + 0 + 0 + 0 + 0 + \cdots + 1 + 0 + 1 + 0 = 13$$

Perhitungan *kernel polynomial* kedua dengan data latih ke-1 dan data ke-2 yang ditunjukan pada Tabel 4.4.

Tabel 4.4 Data Latih Ke-2

Data ke-	G1	G2	G3	G4	G5	 G85	G86	G87	G88
2	1	0	0	0	0	 1	0	1	0

$$k(x,y), \, {\rm misal} \, x = x_1, y = x_2 \, , {\rm maka} \, k(x,y) \colon$$
 
$$k(x_1,x_2) = (x_1,x_2)^2 = (8)^2 = 64$$
 
$$(x_1,x_2) = (1.1) + (0.0) + (0.0) + (0.0) + (0.0) + \cdots + (1.1) + (0.0) + (0.0)$$
 
$$= 1 + 0 + 0 + 0 + 0 + \cdots + 1 + 0 + 1 + 0 = 8$$

Perhitungan *kernel polynomial* kedua dengan data latih ke-1 dan data ke-3 yang ditunjukan pada Tabel 4.5.

Tabel 4.5 Data Latih Ke-3

Data ke-	G1	G2	G3	G4	G5	 G85	G86	G87	G88
3	1	0	0	0	0	 0	1	0	1

$$k(x,y), \, {\rm misal} \, x = x_1, y = x_3 \, , {\rm maka} \, k(x,y) = \\ k(x_1.x_3) = (x_1.x_3)^2 = (1)^2 = 1 \\ (x_1.x_3) = (1.1) + (0.0) + (0.0) + (0.0) + (0.0) + \dots + (1.0) + (0.1) + (0.1) \\ + (0.1) \\ = 1 + 0 + 0 + 0 + 0 + \dots + 0 + 0 + 0 + 0 = 1$$

Seluruh hasil perhitungan pada kernel polynomial dari  $k(x_1, x_1)$  hingga  $k(x_5, x_5)$  ditampilkan pada Tabel 4.6.

Tabel 4.6 Hasil Perhitungan Kernel Polynomial Level 1

k(x,y)	1	2	3	4	5
1	169	64	1	49	0
2	64	144	1	49	0
3	1	1	121	1	81
4	49	49	1	100	0
5	0	0	81	0	121

Tahap selanjutnya setelah menghitung semua nilai pada *kernel* yaitu melakukan tahap *sequential training* SVM. Perhitungan manualisasi pada tahap *training* dijelaskan dibawah ini.

- 1. Inisialisasi nilai  $\alpha_i = 0$ .
- 2. Perhitungan nilai matriks hessian dengan Persamaan 2.10.

 $D_{ij}=y_iy_j(K(x_ix_j)+\lambda^2)$ , dengan  $y_i=1$  menjelaskan kelas pada data kesatu dan  $y_j=1$  menjelaskan kelas pada data kedua, untuk variabel  $(x_ix_j)$  adalah nilai pada kernel. Nilai variabel  $\lambda=0.5$ . Dibawah ini dijelaskan contoh perhitungan matriks hessian untuk data ke-1 dan data ke-2.

$$D_{11} = y_1 y_1 (K(x_1 x_1) + \lambda^2)$$

$$= 1.1(169 + (0.5)^2)$$

$$= 1(169 + 0.25)$$

$$= 169.25$$

$$D_{12} = y_1 y_2 (K(x_1 x_2) + \lambda^2)$$

$$= 1. (-1)(169 + (0.5)^2)$$

$$= 1(64 + 0.25)$$

$$= 64.25$$

Semua perhitungan matriks *hessian* pada perhitungan manual disajikan pada Tabel 4.7.

 Dij
 1
 2
 3
 4
 5

 1
 169,25
 -64,25
 -1,25
 -49,25
 -0,25

Tabel 4.7 Hasil Perhitungan Matriks Hessian Level 1

1	169,25	-64,25	-1,25	-49,25	-0,25
2	-64,25	144,25	1,25	49,25	0,25
3	-1,25	1,25	121,25	1,25	81,25
4	-49,25	49,25	1,25	100,25	0,25
5	-0,25	0,25	81,25	0,25	121,25

3. Pada tahap selanjutnya akan dilakukan perhitungan nilai *error* dengan Persamaan 2.11. Proses dalam menghitung nilai *error* akan dilakukan iterasi hingga *itermax* = 100, *C* (*Complexcity*) = 1, dan variabel γ (*learning rate*) dengan menghitung nilai konstanta = 0.01 dibagi dengan nilai maksimal pada data matriks *hessian*.

$$\gamma = \frac{0.01}{\max D_{ij}}$$
$$\gamma = \frac{0.01}{169.25} = 0.00005908$$

Iterasi ke 1:

a. Perhitungan nilai error pertama yaitu untuk perhitungan data ke-1, dengan inisialisasi awal nilai  $\alpha_i=0$  yaitu seperti dibawah ini:

$$E_i = ((0 * 169.25) + (0 * (-64.25)) + (0 * (-1.25)) + (0 * (-49.25)) + (0 * (-0.25)))$$

$$E_i = 0$$

Semua perhitungan matriks *hessian* pada perhitungan manual disajikan pada Tabel 4.8.

Tabel 4.8 Hasil Perhitungan Nilai  $E_i$  Level 1 Iterasi ke-1

Ei	1	2	3	4	5	Ei
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0

b. Setelah dilakukan perhitungan untuk mencari nilai  $\delta \alpha_i$  dengan Persamaan 2.12. Dibawah ini menampilkan perhitungan pada data ke-1.

$$\delta \alpha_i = \min(\max(\gamma(1 - E_i), -\alpha_i), C - \alpha_i)$$

$$\delta \alpha_i = \min(\max(0.00005908(1-0), -0), 1-0)$$

$$\delta \alpha_i = \min(\max(0.00005908(1), -0), 1)$$

$$\delta \alpha_i = \min(\max(0.00005908, -0), 1)$$

$$\delta \alpha_i = \min(\max(0.00005908), 1)$$

$$\delta \alpha_i = 0.00005908$$

Hasil dari perhitungan untuk nilai  $\delta \alpha_i$  pada semua data latih ditampilkan pada Tabel 4.9.

Tabel 4.9 Hasil Perhitungan Nilai  $\delta lpha_i$  Level 1 Iterasi ke-1

	1	2	3	4	5
$\delta \alpha_1$	5,90842E-05	5,91E-05	5,90842E-05	5,91E-05	5,91E-05

c. Tahap selanjutnya menghitung nilai  $\alpha_i$  baru dengan menggunakan Persamaan 2.13. Dibawah ini menampilkan perhitungan pada data ke-1.

$$\alpha_i = 0 + 0.00005908$$

Hasil dari perhitungan untuk nilai  $\alpha_i$  pada semua data latih ditampilkan pada Tabel 4.10.

Tabel 4.10 Hasil Perhitungan Nilai  $\alpha_i$  Level 1 Iterasi ke-1

	1	2	3	4	5
$\alpha_i$	5,90842E-05	5,91E-05	5,90842E-05	5,91E-05	5,91E-05

d. Setelah mendapat nilai  $\alpha_i$  baru, maka akan dilakukan proses perhitungan untuk iterasi 2. Pada iterasi ke-2 nilai  $\alpha_i$  yang digunakan

adalah nilai  $\alpha_i$  baru yang didapatkan pada iterasi sebelumnya. Pada Tabel 4.11 ditampilkan hasil perhitungan untuk nilai  $E_i$  pada iterasi ke-2.

Tabel 4.11 Hasil Perhitungan Nilai  $E_i$  Level 1 Iterasi ke-2

	1	2	3	4	5	Ei
1	0,01	-0,0038	-7,4E-05	-0,00291	-1,5E-05	0,00321
2	-0,0038	0,00852	7,4E-05	0,00291	1,5E-05	0,00773
3	-7,4E-05	7,4E-05	0,00716	7,4E-05	0,0048	0,01204
4	-0,00291	0,00291	7,4E-05	0,00592	1,5E-05	0,00601
5	-1,5E-05	1,5E-05	0,0048	1,5E-05	0,00716	0,01198

e. Hasil dari perhitungan untuk nilai  $\delta \alpha_i$  untuk iterasi 2 pada keseluruhan data latih ditampilkan pada Tabel 4.12.

Tabel 4.12 Hasil Perhitungan Nilai  $\delta \alpha_i$  Level 1 Iterasi ke-2

	1	2	3	4	5
$\delta lpha_1$	5,9E-05	5,9E-05	5,8E-05	5,9E-05	5,8E-05

f. Hasil dari perhitungan untuk nilai  $\alpha_i$  yang baru pada iterasi ke 2 pada semua data latih ditampilkan pada Tabel 4.13.

Tabel 4.13 Hasil Perhitungan Nilai  $lpha_i$  Level 1 Iterasi ke-2

	1	2	3	4	5
$\alpha_i$	0,00012	0,00012	0,00012	0,00012	0,00012

g. Iterasi akan berhenti ketika kondisi iterasi maksimum (*itermax*) tercapai atau dengan kondisi max ( $|\delta \alpha_i|$ ) <  $\varepsilon$  (*epsilon*).

Jika telah mendapatkan nilai *support vector* pad proses pelatihan data latih sebelumnya, maka proses selanjutnya yaitu melakukan perhitungan untuk mendapatkan nilai w (bobot *vector*) dan nilai b (bias). Variabel  $w.x^+$  menunjukan nilai bobot *support vector* pada kelas positif atau kelas 1, sedangkan Variabel  $w.x^-$  menunjukan nilai bobot *support vector* pada kelas negatif atau kelas -1. Pada level 1 kelas positif terdiri dari kelas 1, dan kelas negatif terdiri dari kelas 2, 3, 4, 5.

Sebelum masuk dalam perhitungan nilai w, akan dilakukan perhitungan nilai  $K(x_i.x^+)$  dan nilai  $K(x_i.x^-)$ . Nilai  $K(x_i.x^+)$  merupakan data kernel pada kelas positif yang didapatkan dari nilai  $\alpha_i$  paling besar pada kelas positif, sedangkan nilai  $K(x_i.x^-)$  merupakan data kernel pada kelas negatif yang didapatkan dari nilai  $\alpha_i$  paling besar pada kelas negatif. Pada Tabel 4.14 ditampilkan nilai kesuluruhan pada nilai  $K(x_i.x^+)$  dan nilai  $K(x_i.x^-)$ .

Tabel 4.14 Hasil Perhitungan  $K(x_i, x^+)$  dan  $K(x_i, x^-)$  Level 1

ID	$K(x_i, x^+)$	$K(x_i, x^-)$
1	169	49
2	64	49
3	1	1
4	49	100
5	0	0

Pada tahapan selanjutnya yaitu melalukan perhitungan nilai  $w.x^+$  dan nilai  $w.x^-$  dengan Persamaan 2.4. Dibawah ini akan diberikan contoh perhitungan pada nilai bobot *vector* kelas positif dan kelas negatif.

$$w. x^{+} = \sum_{i=1}^{n} \alpha_{i} y_{i} k(x_{i}, x^{+})$$

$$w. x^{+} = ((0.00012 * 1 * 169) + (0.00012 * (-1) * 64) + (0.00012 * (-1) * 1) + (0.00012 * (-1) * 49) + (0.00012 * (-1) * 0)$$

$$w. x^{+} = 0.006514584$$

$$w. x^{-} = \sum_{i=1}^{n} \alpha_{i} y_{i} k(x_{i}, x^{-})$$

$$w. x^{-} = ((0.00012 * 1 * 49) + (0.00012 * (-1) * 49) + (0.00012 * (-1) * 1) + (0.00012 * (-1) * 100) + (0.00012 * (-1) * 0)$$

$$w. x^{-} = -0.01188569$$

Perhitungan untuk nilai  $w.x^+$  dan nilai  $w.x^-$  secara keseluruhan ditampilkan pada Tabel 4.15.

Tabel 4.15 Hasil Perhitungan  $w. x^+$  dan  $w. x^-$  Level 1

ID	w. <i>x</i> <sup>+</sup>	w.x <sup>-</sup>
1	0,01994	0,00578
2	-0,00753	-0,00577
3	-0,00012	-0,00012
4	-0,00577	-0,01178
5	0	0
Σ	0,00651	-0,01189

Tahap berikutnya yaitu menghitung nilai *b* (bias) pada level 1 dengan Persamaan 2.3.

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

$$b = -\frac{1}{2}(0.00651 + (-0.01189)) = 0.002685553$$

Tahapan berikutnya adalah melakukan perhitungan pada level 2. Pada proses perhitungan pada level 2, kelas 2 yaitu kelas hebrefenik yang sebelumnya masuk kedalam kelas -1 akan menjadi kelas 1. Untuk kelas 3, 4, dan 5 tetap masuk kedalam kelas -1. Data latih untuk perhitungan level 2 ditampilkan pada Tabel 4.16.

Tabel 4.16 Data Latih Level 2

							Fungsi	Kogniti	f		
NO	Peri	laku Te	rhadap	Peme	riksa			Atens	si dan	Level	Kelas
INO						Mer	nori	Konse	entrasi	Level	Keias
	G1	G2	G3	G4	G5	 G85	G86	G87	G88		
2	1	0	0	0	0	 1	0	1	0	1	2
3	1	0	0	0	0	 0	1	0	1	-1	3
4	1	0	0	0	0	 1	0	0	0	-1	4
5	0	1	0	0	0	 0	1	0	1	-1	5

Proses perhitungan pada level 2 dilakukan sama seperti proses perhitungan pada level 1. Perhitungan dimulai dari menghitung nilai *kernel* hingga mendapat nilai *b* pada level 2. Hasil perhitungan nilai *kernel* pada level 2 ditampilkan pada Tabel 4.17.

Tabel 4.17 Hasil Perhitungan Kernel Polynomial Level 2

k(x,y)	2	3	4	5
2	144	1	49	0
3	1	121	1	81
4	49	1	100	0
5	0	81	0	121

Proses selanjutnya adalah menghitung nilai matriks *hessian*. Hasil perhitungan matriks *hessian* pada level 2 ditampilkan pada Tabel 4.18.

Tabel 4.18 Hasil Perhitungan Matriks Hessian Level 2

Dij	2	3	4	5
2	144,25	-1,25	-49,25	-0,25
3	-1,25	121,25	1,25	81,25
4	-49,25	1,25	100,25	0,25
5	-0,25	81,25	0,25	121,25

Berikutnya adalah menghitung nilai  $E_i$  pada level 2. Hasil perhitungan nilai  $E_i$  pada level 2 ditampilkan pada Tabel 4.19.

Tabel 4.19 Hasil Perhitungan Nilai  $E_i$  Level 2 Iterasi ke-1

$E_i$	2	3	4	5	$E_i$
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Setelah itu dilakukan perhitungan mencari nilai  $\delta\alpha_i$  dengan nilai  $\gamma$  = 0.0000693241. Hasil perhitungan nilai  $\delta\alpha_i$  pada level 2 ditampilkan pada Tabel 4.20.

Tabel 4.20 Hasil Perhitungan Nilai  $\deltalpha_i$  Level 2 Iterasi ke-1

	2	3	4	5
$\delta \alpha_1$	6,93241E-05	6,93241E-05	6,93241E-05	6,93241E-05

Berikutnya mencari nilai  $\alpha_i$ . Hasil perhitungan nilai  $\alpha_i$  pada level 2 ditampilkan pada Tabel 4.21.

Tabel 4.21 Hasil Perhitungan Nilai  $\alpha_i$  Level 2 Iterasi ke-1

	2	3	4	5
$\alpha_i$	6,93241E-05	6,93241E-05	6,93241E-05	6,93241E-05

Setelah mendapat nilai  $\alpha_i$  maka akan dilakukan iterasi kedua. Nilai  $\alpha_i$  yang digunakan pada iterasi kedua akan menggunakan nilai  $\alpha_i$  yang baru. Hasil perhitungan nilai  $E_i$  pada iterasi kedua ditampilkan pada Tabel 4.22.

Tabel 4.22 Hasil Perhitungan Nilai  $E_i$  Level 2 Iterasi ke-2

	2	3	4	5	Ei
2	0,01	-8,66551E-05	-0,003414211	-1,7331E-05	0,006482
3	-8,66551E-05	0,008405546	8,66551E-05	0,005632582	0,014038
4	-0,003414211	8,66551E-05	0,00694974	1,7331E-05	0,00364
5	-1,7331E-05	0,005632582	1,7331E-05	0,008405546	0,014038

Lalu dilakukan perhitungan nilai  $\delta \alpha_i$  dan nilai  $\alpha_i$  pada iterasi kedua. Hasil perhitungan nilai  $\delta \alpha_i$  dan nilai  $\alpha_i$  pada iterasi kedua ditampilkan pada Tabel 4.23 dan Tabel 4.24.

Tabel 4.23 Hasil Perhitungan Nilai  $\delta lpha_i$  Level 2 Iterasi ke-2

	2	3	4	5
$\delta \alpha_1$	6,88747E-05	6,83509E-05	6,90718E-05	6,83509E-05

Tabel 4.24 Hasil Perhitungan Nilai  $\alpha_i$  Level 2 Iterasi ke-2

	2	3	4	5
$\alpha_i$	0,000138199	0,000137675	0,000138396	0,000137675

Pada tahap selanjutnya adalah mencari nilai  $K(x_i, x^+)$  dan  $K(x_i, x^-)$  pada level 2. Hasil perhitungan  $K(x_i, x^+)$  dan  $K(x_i, x^-)$  ditampilkan pada Tabel 4.25.

Tabel 4.25 Hasil Perhitungan Nilai  $K(x_i, x^+)$  dan  $K(x_i, x^-)$  Level 2

ID	$K(x_i, x^+)$	$K(x_i, x^-)$
2	144	49
3	1	1
4	49	100
5	0	0

Selanjutnya dilakukan proses perhitungan nilai  $w.x^+$  dan nilai  $w.x^-$  pada level 2. Hasil perhitungan ditampilkan pada Tabel 4.26.

Tabel 4.26 Hasil Perhitungan Nilai  $w. x^+$  dan nilai  $w. x^-$  Level 2

ID	w. <i>x</i> <sup>+</sup>	w. x <sup>-</sup>
2	0,019900632	0,006771743
3	-0,000137675	-0,000137675
4	-0,006781398	-0,013839587
5	0	0
Σ	0,012981559	-0,007205519

Tahap berikutnya mencari nilai b pada level 2 dengan Persamaan 2.3.

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

$$b = -\frac{1}{2}(0.012981559 + (-0.0072005519)) = -0.00288802$$

Setelah mendapatkan nilai *b* pada level 2, langkah selanjutnya adalah melakukan perhitungan level 3 dan level 4. Proses perhitungan pada level 3 dan 4 dilakukan dengan cara yang sama seperti perhitungan pada level 1 dan 2, tetapi pada level 3 kelas yang bernilai 1 adalah kelas 3 yaitu katatonik, sedangkan untuk level 4 kelas yang bernilai 1 adalah kelas 4 yaitu *undifferentiated* dan untuk kelas 5 atau kelas simpleks akan bernilai -1.

#### 4.4.2 Proses Pengujian Data Uji SVM

Dalam proses pengujian ini akan dilakukan perhitungan untuk mencari nilai f(x) dengan Persamaan 2.5. Nilai f(x) digunakan untuk menentukan kelas pada data yang diuji. Data uji yang digunakan pada perhitungan manual adalah sebanyak 2 data, kelas asli pada data ke-6 adalah kelas paranoid dan data ke-7

adalah kelas katatonik. Pada Tabel 4.27 menampilkan data uji yang akan digunakan.

Tabel 4.27 Data Uji

							Fungsi Kognitif				
NO	Peril	Perilaku Terhadap Pemeriksa							Atensi	i dan	Kelas
INO					Mem	nori	Konser	ntrasi	Keias		
	G1	G2	G3	G4	G5		G85	G86	G87	G88	
6	1	0	0	0	0		1	0	1	0	1
7	1	0	0	0	0		0	1	0	1	3

Data ke-6 akan digunakan sebagai contoh perhitungan perngujian secara manual. Langkah awal pada proses pengujian adalah menghitung nilai *kernel* pada data ke-6 dengan semua data latih. Dibawah ini akan ditampilkan contoh perhitungan *kernel* pada data latih ke-1 dengan data uji ke-6.

$$k(x_i,x), \, \mathrm{misal} \, x_i = x_1, x = x_6 \, , \mathrm{maka} \, k(x_i,x) \colon$$
 
$$k(x_1,x_6) = (x_1,x_6)^2 = (10)^2 = 100$$
 
$$(x_1,x_6) = (1.1) + (0.0) + (0.0) + (0.0) + (0.0) + \cdots + (1.1) + (0.0) + (1.1) + (0.0)$$
 
$$= 1 + 0 + 0 + 0 + 0 + \cdots + 1 + 0 + 1 + 0 = 10$$

Lalu perhitungan kernel data uji ke-6 dengan data latih ke-2.

$$k(x_i,x), \, {\rm misal} \, x_i = x_2, x = x_6 \, , {\rm maka}$$
 
$$k(x_i,x) = k(x_2.x_6) = (x_2.x_6)^2 = (7)^2 = 49$$
 
$$(x_2.x_6) = (1.1) + (0.0) + (0.0) + (0.0) + (0.0) + \cdots + (1.1) + (0.0) + (1.1) + (0.0)$$
 
$$= 1 + 0 + 0 + 0 + \cdots + 1 + 0 + 1 + 0 = 7$$

Setelah mendapatkan nilai kernel pada semua data selanjutnya akan dilakukan perhitungan untuk mencari nilai  $\alpha_i y_i K(x_i, x)$ . Berikut ini adalah contoh perhitungannya.

$$\alpha_i y_i K(x_1, x) = 0,000117979 * 1 * 100 = 0.0117979$$

Data ke-2

$$\alpha_i y_i K(x_2, x) = 0.000117712 * (-1) * 49 = -0.005767888$$

Data ke-3

$$\alpha_i y_i K(x_3, x) = 0.000117457 * (-1) * 1 = -0.000117457$$

Data ke-4

$$\alpha_i y_i K(x_4, x) = 0.000117813 * (-1) * 49 = -0.005772837$$

Data ke-5

$$\alpha_i y_i K(x_5, x) = 0.000117461 * (-1) * 0 = 0$$

Untuk hasil keseluruhan pada perhitungan manual sebelumnya ditampilkan pada Tabel 4.28.

Tabel 4.28 Perhitungan nilai *Kernel* dan nilai  $\alpha_i y_i K(x_i, x)$  Data ke-6 Level 1

ID	$K(x_i,x)$	$\alpha_i y_i K(x_i, x)$		
1	100	0,0117979		
2	49	-0,005767888		
3	1	-0,000117457		
4	49	-0,005772837		
5	0	0		
2		0,000139718		

Tahap berikutnya yaitu mencari nilai f(x) dan nilai fungsi klasifikasi. Nilai b (bias) yang digunakan dalam mencari nilai f(x) adalah nilai b pada level 1, dan setiap perhitungan f(x) pada level berikutnya nilai b akan sesuasi dengan nilai b pada masing-masing level.

$$f(x) = \sum_{i=1}^{m} \alpha_i y_i x_i \cdot x + b$$
 
$$f(x) = 0.000139718 + 0.002685553 = 0.002825271$$
 Fungsi Klasifikasi =  $sign(0.002825271) = 1$ 

Setelah mendapatkan nilai fungsi klasifikasi, dan hasilnya adalah 1 maka data ke-6 merupakan kelas *paranoid*. Penjelasan mengenai kelas pada hasil klasifikasi di proses pengujian dengan kelas sebenarnya ditampilkan pada Tabel 4.29.

Tabel 4.29 Hasil Pengujian pada Data ke-6 dan ke-7

ID	$f^1(x)$	$f^2(x)$	$f^3(x)$	$f^4(x)$	Hasil	Kelas
					Klasifikasi	Sebenarnya
6	0.002825271	-	-	-	1 di level 1	Paranoid
					= Paranoid	
7	-0.01869203	-0.027807392	0.004777722	-	1 di level 3	Hebrefenik
					=	
					Hebrefenik	

#### 4.4.3 Perhitungan Nilai Akurasi

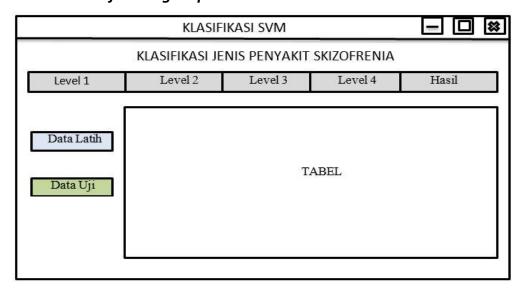
Keberhasilan suatu sistem dalam mengklasifikasikan suatu data dinilai dari tingkat akurasi yang dihasilkan. Dalam mengukur tingkat akurasi suatu sistem dapat dilakukan dengan cara membandingkan banyaknya data yang berhasil diklasifikasikan dengan jumlah data keseluruhan yang diuji. Formulasi yang digunakan dalam melakukan perhitungan nilai akurasi adalah dengan menggunakan Persamaan 2.13.

Nilai Akurasi = 
$$\frac{\text{jumlah h data akurat}}{\text{jumlah h seluruh h data}} * 100\% = \frac{2}{2}x100\% = 100\%$$

### 4.5 Perancangan User Interface

Fungsi perancangan *user interface* adalah untuk memberikan gambaran mengenai *user interface* yang akan dibuat pada sistem klasifikasi jenis penyakit Skizofrenia. Rancangan *interface* pada sistem ini akan menampilkan 7 halaman. Halaman akan terdiri dari halaman data latih, data uji, hasil pengujian, dan hasil perhitungan pada level 1 hingga level 4.

### 4.5.1 User Interface Page Input Data Latih

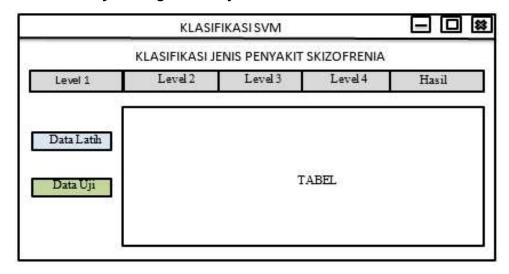


Gambar 4.12 Perancangan Page Input Data Latih

#### Keterangan:

1. Tombol data latih berfungsi untuk memanggil data latih yang digunakan untuk proses perhitungan dalam bentuk .csv.

#### 4.5.2 User Interface Page Data Uji

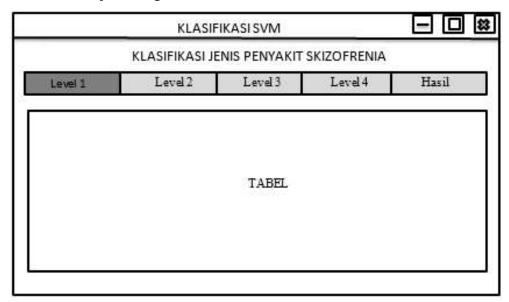


Gambar 4.13 Perancangan Page Input Data Uji

#### Keterangan:

1. Tombol data uji berfungsi untuk memanggil data uji yang digunakan untuk proses perhitungan dalam bentuk .csv

# 4.5.3 User Interface Page Level 1

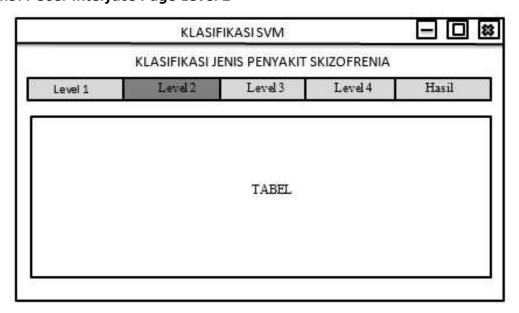


Gambar 4.14 Perancangan Page Level 1

#### Keterangan:

1. Tombol level 1 berfungsi untuk menampilkan semua hasil perhitungan data latih yang ada pada level 1.

# 4.5.4 User Interface Page Level 2

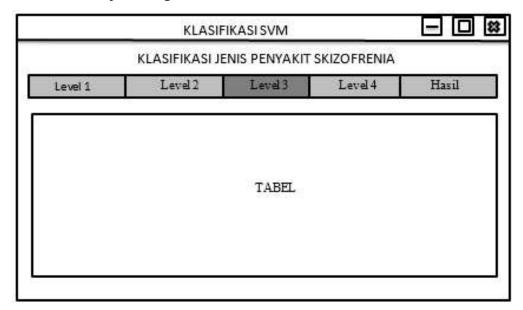


Gambar 4.15 Perancangan Page Level 2

### Keterangan:

1. Tombol level 2 berfungsi untuk menampilkan semua hasil perhitungan data latih yang ada pada level 2.

# 4.5.5 User Interface Page Level 3

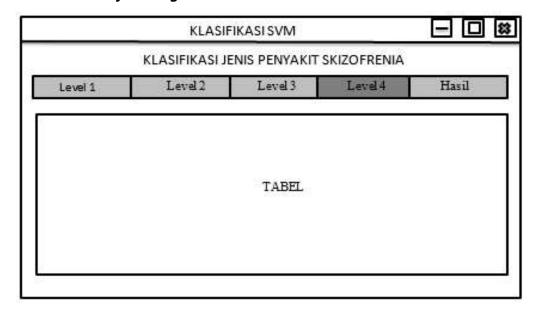


Gambar 4.16 Perancangan Page Level 3

# Keterangan:

1. Tombol level 3 berfungsi untuk menampilkan semua hasil perhitungan data latih yang ada pada level 3.

# 4.5.6 User Interface Page Level 4

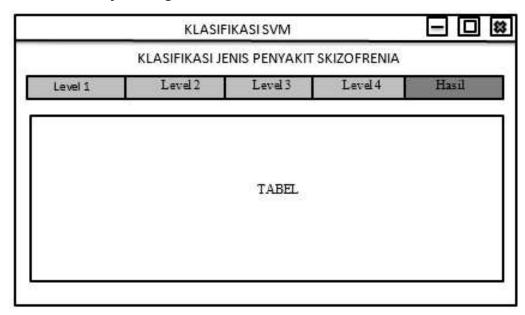


Gambar 4.17 Perancangan Page Level 4

#### Keterangan:

1. Tombol level 4 berfungsi untuk menampilkan semua hasil perhitungan data latih yang ada pada level 4.

### 4.5.7 User Interface Page Hasil



Gambar 4.18 Perancangan Page Hasil

#### Keterangan:

1. Tombol hasil berfungsi untuk menampilkan semua hasil perhitungan data uji. Halaman Hasil akan berisi nilai f(x) dan fungsi klasifikasi pada data yang diuji.

#### 4.6 Perancangan Pengujian Sistem

Perancangan selanjutnya yaitu perancangan pengujian sistem. Pada perancangan ini akan dibuat suatu skenario untuk menguji jenis kernel, variabel  $\lambda$  (Lambda), variabel C (Complexity), variabel  $\gamma$  (Gamma) dan jumlah itermax terhadap tingkat akurasi sistem yang dibuat. Skenario pengujian akan menggunakan jenis pengujian K-fold  $Cross\ Validation$ , dimana jumlah K-fold yang digunakan sebanyak 11 fold.

#### 4.6.1 Perancangan Pengujian Jenis Kernel

Fungsi dari perancangan ini adalah untuk menemukan jenis *kernel* terbaik sebagai perhitungan *kernel* untuk meningkatkan tingkat akurasi sistem. Pada pengujian ini akan diberikan beberapa jenis *kernel* yang berbeda. Tabel 4.30 adalah rancangan pengujian yang digunakan dalam menguji beberapa jenis *kernel*.

Tabel 4.30 Rancangan Pengujian Jenis Kernel

No	Fold ke-i (%)	Jenis Kernel									
INO	1010 KE-1 (70)	Kernel Linier	Kernel Polynomial	Kernel RBF							
1	1										
2	2										
3	3										
4	4										
5	5										
6	6										
7	7										
8	8										
9	9										
10	10										
11	11										
Rata-ı	rata Akurasi (%)										

# **4.6.2** Perancangan Pengujian Variabel *λ* (*Lambda*)

Fungsi dari perancangan ini adalah untuk menemukan nilai variabel  $\lambda$  terbaik untuk meningkatkan tingkat akurasi sistem. Pada pengujian ini akan diberikan beberapa nilai variabel  $\lambda$  yang berbeda. Tabel 4.31 adalah rancangan pengujian yang digunakan dalam menguji beberapa variabel  $\lambda$  yang berbeda.

Tabel 4.31 Rancangan Pengujian Nilai Variabel  $\lambda$ 

No	Fold ke-i (%)		Nilai λ									
100	7 010 KC 7 (70)	0.1	0.5	1	5	10	50	100	500			
1	1											
2	2											
3	3											
4	4											
5	5											
6	6											
7	7											
8	8											
9	9											

10	10				
11	11				
Rata (%)	a-rata Akurasi				

### 4.6.3 Perancangan Pengujian Variabel C (Complexity)

Fungsi dari perancangan ini adalah untuk menemukan nilai variabel *C* terbaik untuk meningkatkan tingkat akurasi sistem. Pada pengujian ini akan diberikan beberapa nilai variabel *C* yang berbeda. Tabel 4.32 adalah rancangan pengujian yang digunakan dalam menguji beberapa variabel *C* yang berbeda.

Tabel 4.32 Rancangan Pengujian Nilai Variabel C

No	Fold ke-i (%)				Nila	ai <i>C</i>			
	7 010 KC 7 (70)	0.1	0.5	1	10	30	50	70	100
1	1								
2	2								
3	3								
4	4								
5	5								
6	6								
7	7								
8	8								
9	9								
10	10								
11	11								
Rata (%)	a-rata Akurasi								

#### 4.6.4 Perancangan Pengujian Nilai Variabel y (Gamma)

Fungsi dari perancangan ini adalah untuk menemukan nilai variabel  $\gamma$  terbaik untuk meningkatkan tingkat akurasi sistem. Pada pengujian ini akan diberikan beberapa nilai variabel  $\gamma$  yang berbeda. Tabel 4.33 adalah rancangan pengujian yang digunakan dalam menguji beberapa variabel  $\gamma$  yang berbeda.

Tabel 4.33 Rancangan Pengujian Nilai Variabel y

No	Fold ke-i (%)				Nilai	γ		
	1 010 KE-1 (70)	0.001	0.01	0.1	0.5	1	5	10
1	1							
2	2							
3	3							
4	4							
5	5							
6	6							
7	7							
8	8							
9	9							
10	10							
11	11							
Rata (%)	i-rata Akurasi							

# 4.6.5 Perancangan Pengujian Jumlah Iterasi

Fungsi dari perancangan ini adalah untuk menemukan jumlah iterasi terbaik untuk meningkatkan tingkat akurasi sistem. Pada pengujian ini akan diberikan beberapa jumlah *itermax* yang berbeda. Tabel 4.34 adalah rancangan pengujian yang digunakan dalam menguji beberapa jumlah iterasi yang berbeda.

Tabel 4.34 Rancangan Pengujian Jumlah Iterasi

No	Fold ke-i	Jumlah Iterasi									
INO	(%)	100	500	1000	5000	10000					
1	1										
2	2										
3	3										
4	4										
5	5										
6	6										
7	7										
8	8										

9	9			
10	10			
11	11			
Rata-rata Akurasi (%)				