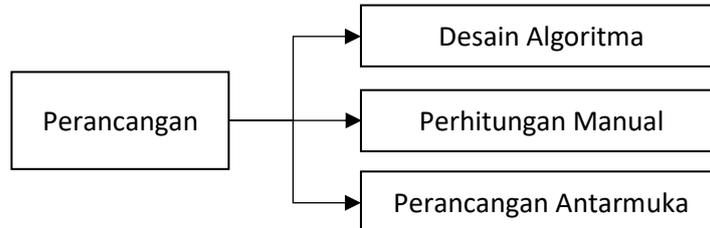


BAB 4 PERANCANGAN

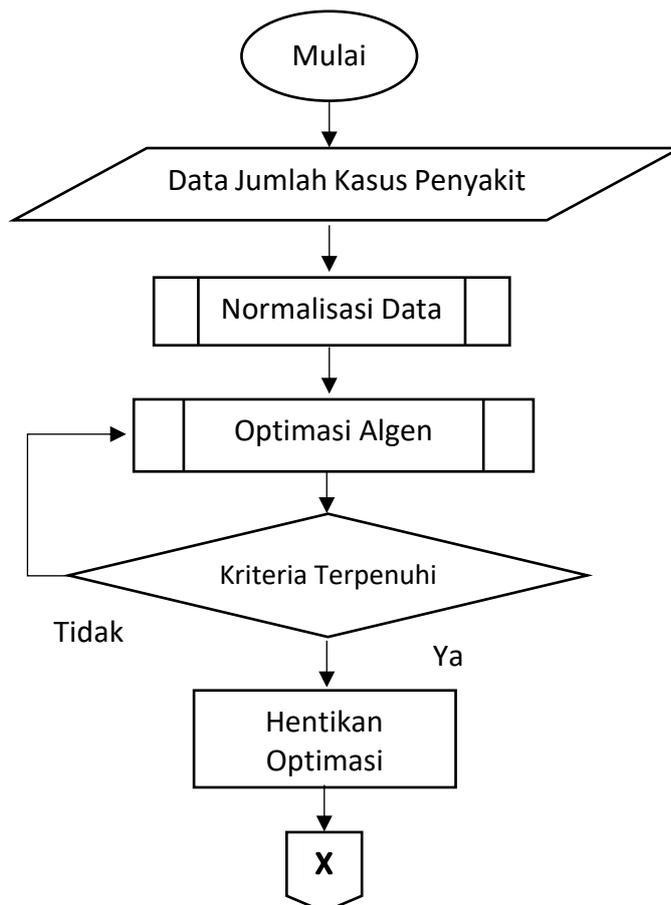
Bab ini akan membahas proses perancangan optimasi peramalan jumlah kasus penyakit menggunakan metode jaringan syaraf tiruan *backpropagation* dengan algoritma genetika. Tahapan dari perancangan dijelaskan dengan Gambar 4.1:

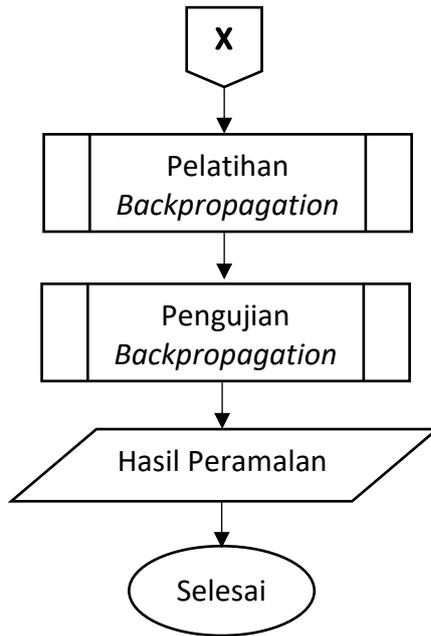


Gambar 4.1 Pohon Perancangan

4.1 Desain Algoritma

4.1.1 Desain Algoritma *Backpropagation* – Algoritma Genetika





Gambar 4.2 Diagram Alir Perancangan Algoritma *Backpropagation* – Algoritma Genetika

4.1.2 Persiapan Data

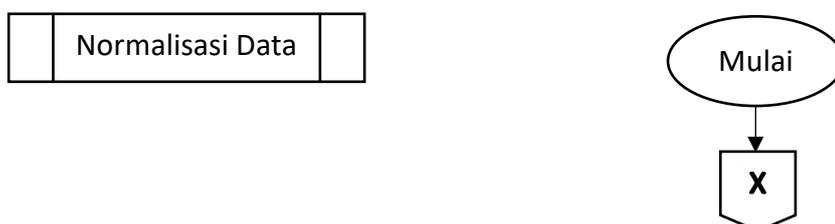
Sebelum melakukan proses peramalan data akan dipersiapkan dengan mengambil data latih jumlah kasus penyakit demam *typhoid-paratyphoid* dari bulan September tahun 2015 sampai bulan Desember tahun 2015 yang dibagi menjadi 4 data lalu menyusunnya sedemikian rupa, ditunjukkan dengan Tabel 4.1:

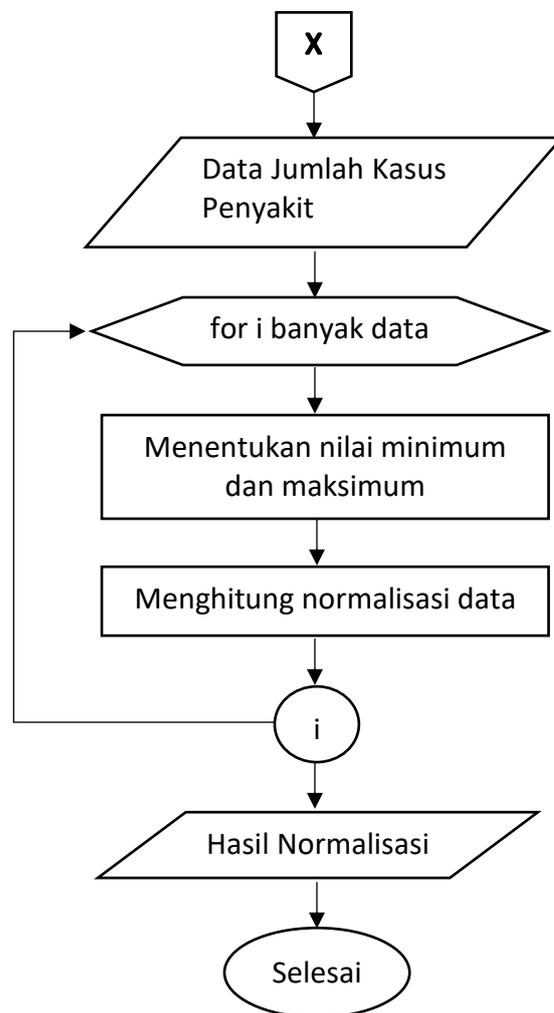
Tabel 4.1 Data Latih Jumlah Kasus Penyakit Demam *Typoid-Paratyphoid*

Data	x1	x2	x3	x4	x5	x6	t
1	37	24	16	11	7	7	8
2	24	16	11	7	7	8	15
3	16	11	7	7	8	15	33
4	11	7	7	8	15	33	20

4.1.3 Normalisasi Data

Normalisasi data digunakan untuk mempermudah proses perhitungan dengan cara mengubah data aktual menjadi data dengan interval 0 sampai 1. Diagram alir dari proses normalisasi data ditunjukkan dengan Gambar 4.1:





Gambar 4.3 Diagram Alir Normalisasi Data

Berdasarkan diagram alir pada Gambar 4.3 terlebih dahulu dilakukan pencarian nilai maksimum serta minimum dari data yang digunakan untuk melakukan proses normalisasi. Berdasarkan Tabel 3.1 nilai maksimum adalah 89 dan nilai minimum adalah 0. Untuk menghitung normalisasi data digunakan persamaan 4.1. Hasil dari normalisasi ditunjukkan pada Tabel 4.2:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} = \frac{31 - 0}{89 - 0} = \frac{31}{89} = 0,34831461 \quad 4.1$$

dimana

x_{max} = nilai maksimum data

x_{min} = nilai minimum data

x = nilai yang dinormalisasi

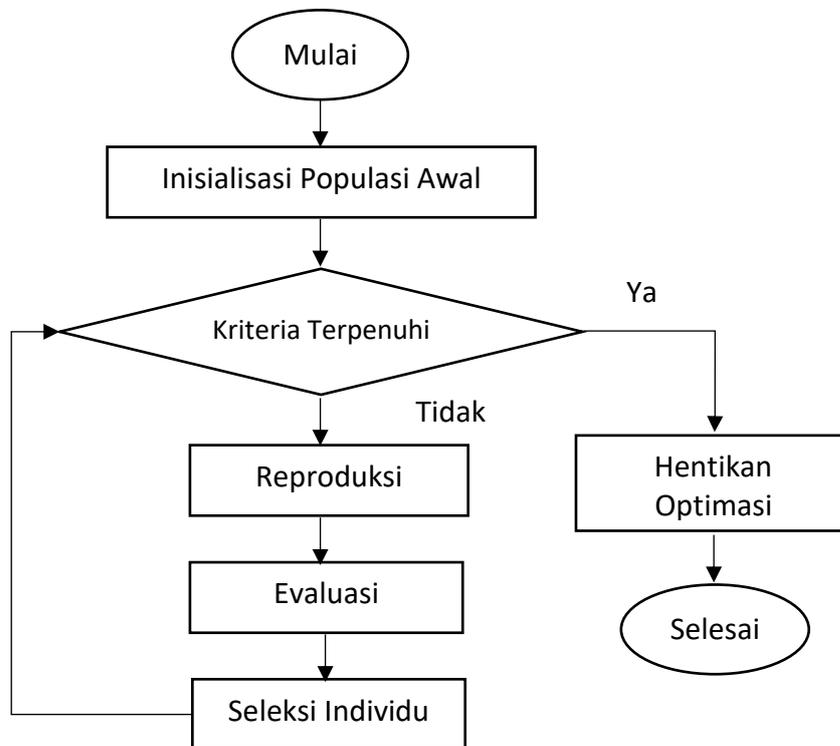
Tabel 4.2 Data Hasil Normalisasi

Data	x1	x2	x3	x4	x5	x6	t
1	1	0.56667	0.3	0.13333	0	0	0.03333
2	0.56667	0.3	0.13333	0	0	0.03333	0.26667
3	0.3	0.13333	0	0	0.03333	0.26667	0.86667
4	0.13333	0	0	0.03333	0.26667	0.86667	0.43333

4.1.4 Optimasi Algoritma Genetika

Sebelum masuk dalam proses peramalan *backpropagation*, akan dilakukan proses optimasi bobot serta bias awal ditunjukkan dengan Gambar 4.4:

	Optimasi Algen	
--	----------------	--



Gambar 4.4 Diagram Alir Optimasi Algoritma Genetika

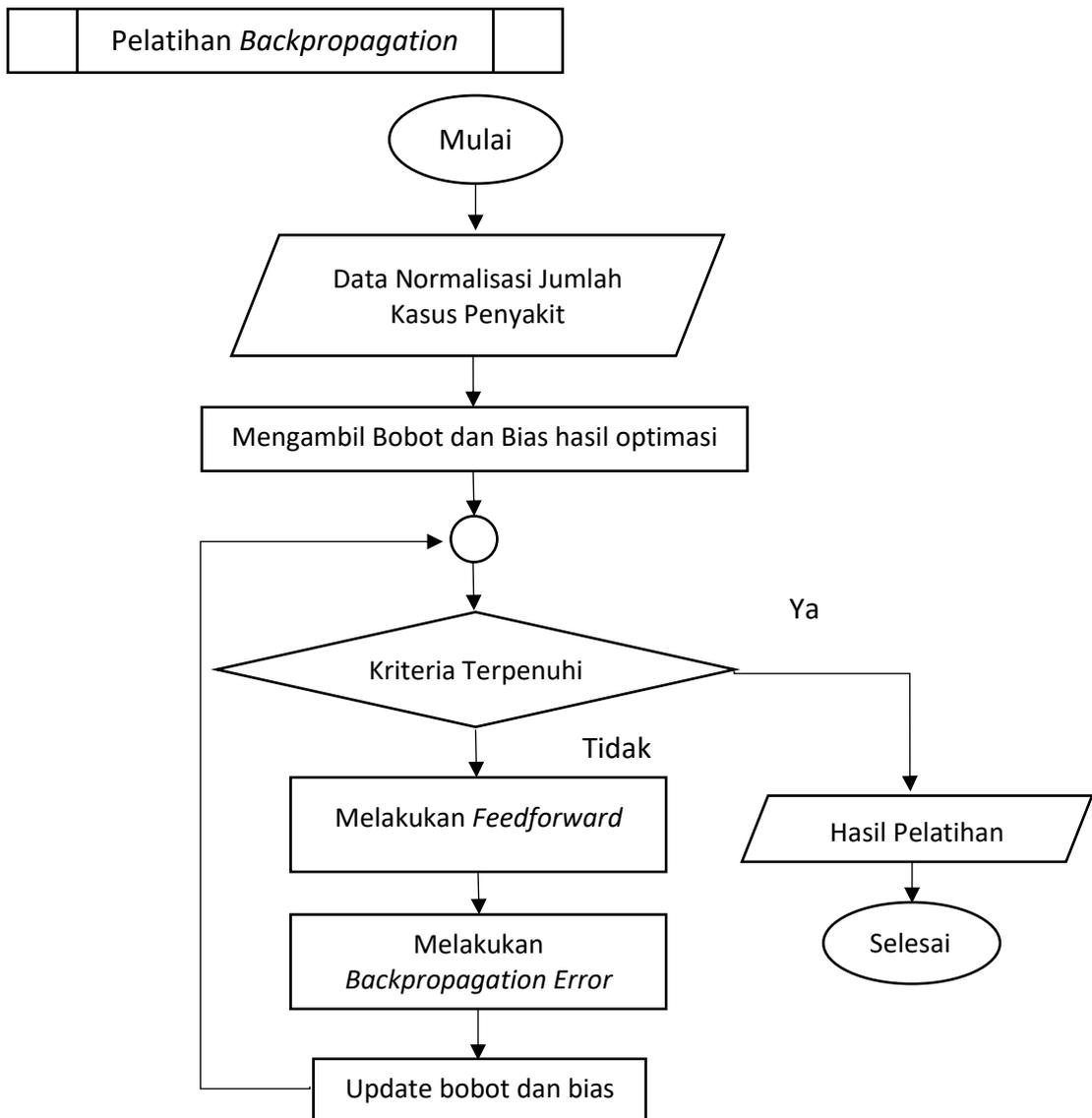
Berdasarkan Gambar 4.4, langkah-langkah untuk melakukan optimasi bobot serta bias v dan w adalah sebagai berikut:

1. Melakukan inisialisasi populasi awal sebanyak *popsiz*e yang ditentukan.
2. Mengecek apakah kriteria berhenti terpenuhi atau belum. Kriteria yang dimaksud adalah jumlah generasi atau iterasi.

3. Menjalankan proses reproduksi yaitu crossover dan mutasi. Crossover menggunakan *one-cut point* serta mutasi menggunakan *random mutation*.
4. Melakukan evaluasi dengan menghitung *fitness* menggunakan *feedforward*.
5. Melakukan proses seleksi dengan metode *elitism selection*.

4.1.5 Pelatihan *Backpropagation*

Untuk mendapatkan hasil peramalan jumlah kasus penyakit maka dilakukan proses pelatihan *backpropagation*. Tahapan pelatihan *backpropagation* ditunjukkan dengan Gambar 4.5:



Gambar 4.5 Diagram Alir Pelatihan *Backpropagation*

Terdapat beberapa parameter yang digunakan pada perhitungan manual yang ditunjukkan Tabel 4.3:

Tabel 4.3 Parameter Backpropagation

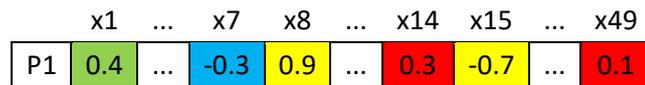
Epoch	α	n
2	0.1	4

4.2 Perhitungan Manual

Pada perhitungan manual digunakan data latih untuk meramalkan data pada bulan Juli 2016. Setelah melakukan proses normalisasi data selanjutnya dilakukan proses optimasi dengan menggunakan algoritma genetika.

4.2.1 Inisialisasi Populasi Awal

Langkah pertama yaitu melakukan inisialisasi populasi awal sebanyak *popsiz*e yang digunakan yaitu 5. Setiap individu yang di inisialisasi merepresentasikan bobot serta bias w dan v . Panjang *chromosome* untuk setiap individu adalah 49, dimana posisi x_1 - x_7 merupakan bobot serta bias dari w dan posisi x_8 - x_{49} merupakan bobot serta bias dari v . Nilai dari *chromosome* ini didapatkan dari hasil random antara -0,1 sampai 0,9. Representasi *chromose* ditunjukkan dengan Gambar 4.6:



Gambar 4.6 Representasi Chromosome

Keterangan:

Hijau : Bobot w

Biru : Bias w

Kuning : Bobot v

Merah : Bias v

Setelah melakukan inisialisasi populasi, kemudian dilakukan proses perhitungan nilai *fitness* menggunakan Persamaan 4.2:

$$fitness = \frac{1}{MSE} \quad 4.2$$

Hasil inisialisasi populasi awal serta *fitness* nya ditunjukkan dengan Tabel 4.4:

Tabel 4.4 Inisialisasi Populasi Awal

	Chromosome							<i>fitness</i>
	x1	x2	x3	x4	x5	x_i	x49	
P1	0.4	-0.2	0.5	-0.2	-0.6	...	0.1	0.009796
P2	0.7	-0.5	0.4	-0.3	0.6	...	-0.5	0.011808
P3	0.2	-0.8	-0.2	-0.9	-0.9	...	-0.3	0.006726
P4	0.6	0.7	-0.1	-0.4	0.5	...	-0.3	0.00599
P5	0.1	0.5	0.6	-0.8	0.8	...	0.1	0.013481

4.2.2 Reproduksi

Setelah mendapatkan populasi selanjutnya dilakukan proses reproduksi yaitu *crossover* serta mutasi. Metode yang digunakan untuk proses *crossover* adalah *one-cut point*, yaitu dengan cara memotong kemudian menukar gen dengan titik potong acak dari 1 hingga panjangnya *chromosome*. Pada proses perhitungan manual ini digunakan 2 titik potong yaitu 4 untuk bobot serta bias w dan 12 untuk bobot serta bias v dengan induk 3 (P3) dan induk 5(P5) ditunjukkan dengan Gambar 4.7.

	Cut Point 1				Cut Point 2									
P3	0.2	-0.8	-0.2	-0.9	-0.9	-0.8	-0.1	0.4	...	0.7	0.1	...	-0.3	
P5	0.1	0.5	0.6	-0.8	0.8	-0.6	-0.9	0.7	...	0.7	0.2	...	0.1	
	x1-x4				x5-x7			x8-x19			x20-49			

Gambar 4.7 Proses Crossover

Untuk proses mutasi menggunakan metode *random mutation* dengan menggunakan persamaan 2.25.

$$x'_i = x_i + r(max_i - min_i)$$

dimana

x_i = gen yang terpilih untuk mutasi

r = nilai random -0,1 sampai 0,1

max_i = nilai maksimum dari individu yang terpilih untuk mutasi

min_i = nilai minimum dari individu yang terpilih untuk mutasi

Crossover rate (cr) dan *mutation rate* (mr) yang digunakan dalam perhitungan manual ini berturut-turut yaitu 0.01 dan 0.2. Individu yang terpilih untuk *crossover* adalah P3 dan P5 serta mutasi adalah P4. Berdasarkan cr dan mr yang ditetapkan, didapat *offspring* sebanyak 2 untuk masing-masing *crossover* dan mutasi. Setelah melakukan reproduksi maka didapatkan keturunan atau *child* yaitu C1 dari *crossover* dan C2 dari mutasi. Berikut adalah hasil *crossover* dan mutasi ditunjukkan dengan Gambar 4.8:

C1	0.2	-0.8	-0.2	-0.9	0.8	...	0.1
C2	0.6	0.7	-0.1	-0.4	0.5	...	-0.3

Gambar 4.8 Keturunan Hasil Reproduksi

4.2.3 Evaluasi

Setelah mendapatkan keturunan dilanjutkan dengan melakukan proses evaluasi untuk menghitung nilai *fitness* dari individu hasil reproduksi menggunakan nilai MSE yang didapat dari proses *feeforward*.

Langkah pertama dilakukan inialisasi bobot serta bias dari hasil reproduksi. Pada perhitungan manual diambil contoh menghitung nilai *fitness* dari C1. Pada

Tabel 4.5 dan 4.6, v, x1-x6 dan z1-z6 melambangkan bobot serta bias untuk layer input ke layer tersembunyi dan w, z1-z6, y melambangkan bobot serta bias untuk layer tersembunyi ke layer output.

Tabel 4.5 Bobot serta bias v

v	x1	x2	x3	x4	x5	x6	1
z1	0.4	0.2	-0.7	0.4	-0.9	0.5	0.2
z2	0.6	-0.4	-0.9	-0.9	0.7	0.2	0.8
z3	-0.6	0.4	-0.3	0.6	-0.7	0.7	0.2
z4	0.1	0.8	0.8	0.1	0.2	0.3	-0.4
z5	-0.8	-0.9	-0.8	-0.9	0.8	0.7	-0.5
z6	-0.6	-0.6	-0.3	-0.4	0.4	-0.1	0.1

Tabel 4.6 Bobot serta bias w

w	y
z1	0.2
z2	-0.8
z3	-0.2
z4	-0.9
z5	0.8
z6	-0.6
1	-0.9

Setelah mendapatkan bobot serta bias v dan w maka dilanjutkan ke proses *feedforward*. Digunakan data normalisasi seperti pada Tabel 4.2 sebagai data training. Langkah-langkah proses *feedforward* adalah menghitung nilai z_{in} untuk data pertama. Nilai z_{in} yang dihitung adalah sebanyak unit bobot v yang ada yaitu 6. Nilai z_{in} dihitung menggunakan Persamaan 2.10. Hasil z_{in} ditunjukkan dengan Tabel 4.7.

$$z_{in1} = v_{01} + \sum_{i=1}^n x_i v_{i1}$$

$$z_{in1} = v_{01} + (x_1 v_{11}) + (x_2 v_{21}) + \dots + (x_6 v_{61})$$

$$z_{in1} = 0.2 + (0.4) + (0.113333333) + (-0.21) + (0.053333333) + (0) + (0)$$

$$z_{in1} = \mathbf{0.556666667}$$

Tabel 4.7 Nilai z_{in}

z_in1	0.556666667
z_in2	0.783333333

z_in3	-0.183333333
z_in4	0.406666667
z_in5	-2.17
z_in6	-0.983333333

Langkah selanjutnya adalah menghitung nilai z dengan cara memasukkan nilai z_{in} kedalam fungsi aktivasi yaitu *binary* sigmoid. Nilai z dihitung dengan Persamaan 2.11. Hasil z ditunjukkan dengan Tabel 4.8.

$$z_1 = f(z_{in1}) = \frac{1}{1 + \exp^{-(z_{in1})}}$$

$$z_1 = \mathbf{0.635680921}$$

Tabel 4.8 Nilai z

z1	0.635680921
z2	0.686398079
z3	0.454294612
z4	0.600288339
z5	0.102477033
z6	0.272230878

Selanjutnya menghitung nilai y_{in} . Karena hanya terdapat satu *output* maka hanya terdapat satu unit y_{in} . Nilai y_{in} dihitung dengan persamaan 2.12. Nilai y_{in} yang didapat yaitu:

$$y_{in} = w_0 + \sum_{i=1}^n z_i w_i$$

$$y_{in} = w_0 + (z_1 w_1) + (z_2 w_2) + \dots + (z_6 w_6)$$

$$y_{in} = -0.9 + (0.127136184) + (-0.549118463) + (-0.090858922) + (-0.540259505) + (0.081981627) + (-0.163338527)$$

$$y_{in} = \mathbf{-2.034457607}$$

Selanjutnya adalah menghitung nilai y menggunakan fungsi aktivasi pada Persamaan 2.13. Nilai y yang didapat yaitu:

$$y = f(y_{in}) = \frac{1}{1 + e^{-(y_{in})}}$$

$$y = \mathbf{0.115632299}$$

Selanjutnya proses ini diulang sebanyak data latih yang digunakan yaitu empat. Kemudian setelah nilai y didapatkan, proses selanjutnya adalah melakukan *denormalisasi* nilai y untuk menghitung nilai MSE menggunakan Persamaan 4.3 dengan data latih.

$$x = x'(x_{max} - x_{min}) + x_{min}$$

$$y = 0.115632299 \times (37 - 7) + 7$$

$$y = \mathbf{10.46896897}$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (t_k - y_k)^2$$

$$MSE = \frac{((8-10.46)^2 + (15-10.69)^2 + (33-10.95)^2 + (20-11.17)^2)}{4}$$

$$MSE = \mathbf{147.1091206}$$

Selanjutnya menghitung nilai *fitness* dengan menggunakan Persamaan 4.2. Nilai *fitness* yang didapat yaitu:

$$fitness = \frac{1}{MSE}$$

$$fitness = \frac{1}{147.1091206}$$

$$fitness = \mathbf{0.006797675}$$

Hasil evaluasi ditunjukkan dengan Tabel 4.9:

Tabel 4.9 Hasil Evaluasi

	Chromosome							<i>fitness</i>
	x1	x2	x3	x4	x5	xi	x49	
P1	0.4	-0.2	0.5	-0.2	-0.6	...	0.1	0.009795541
P2	0.7	-0.5	0.4	-0.3	0.6	...	-0.5	0.011808249
P3	0.2	-0.8	-0.2	-0.9	-0.9	...	-0.3	0.006726128
P4	0.6	0.7	-0.1	-0.4	0.5	...	-0.3	0.005990015
P5	0.1	0.5	0.6	-0.8	0.8	...	0.1	0.013480802
C1	0.2	-0.8	-0.2	-0.9	0.8	...	0.1	0.006797675
C2	0.6	0.7	-0.1	-0.4	0.5	...	-0.3	0.006029822

4.2.4 Seleksi

Selanjutnya dilakukan proses seleksi individu dengan menggunakan *elitism selection* dengan memilih individu terbaik sebanyak *popsiz*, yaitu 5 individu. Setelah proses seleksi maka akan didapatkan populasi atau generasi selanjutnya yang ditunjukkan dengan Tabel 4.10. Dari tabel tersebut didapatkan individu terbaik yaitu P1 yang memiliki nilai *fitness* sebesar 834.5254834. Pada perhitungan manual ini proses optimasi dilakukan sebanyak 2 generasi.

Tabel 4.10 Generasi Baru ke 2

P(t+1)	Asal	Chromosome							<i>fitness</i>
		x1	x2	x3	x4	x5	xi	x49	
P1	P1	0.1	0.5	0.6	-0.8	0.8	...	0.1	0.013480802

P2	P2	0.7	-0.5	0.4	-0.3	0.6	...	-0.5	0.011808249
P3	C1	0.4	-0.2	0.5	-0.2	-0.6	...	0.1	0.009795541
P4	P3	0.2	-0.8	-0.2	-0.9	0.8	...	0.1	0.006797675
P5	C2	0.2	-0.8	-0.2	-0.9	-0.9	...	-0.3	0.006726128

Dari proses optimasi algoritma genetika didapatkan nilai bobot serta bias v dan w yang optimal. Bobot dan bias tersebut akan digunakan pada proses algoritma *backpropagation*.

4.2.5 Proses *Feedforward*

Setelah mendapat bobot optimal hasil optimasi, langkah selanjutnya adalah proses pelatihan algoritma *backpropagation* yang dimulai dengan proses *feedforward*. Langkah pertama adalah mendapatkan bobot dan bias optimal dari hasil optimasi. Bobot dan bias optimal ditunjukkan dengan Tabel 4.11 dan 4.12.

Tabel 4.11 Bobot serta Bias v

v	z_1	z_2	z_3	z_4	z_5	z_6
x_1	0.7	-0.2	0.9	0.4	0.3	-0.5
x_2	0.7	-0.6	-0.5	0.2	0.7	0.2
x_3	-0.6	0.4	-0.3	0.6	-0.7	0.7
x_4	0.1	0.8	0.8	0.1	0.2	0.3
x_5	-0.8	-0.9	-0.8	-0.9	0.8	0.7
x_6	-0.6	-0.6	-0.3	-0.4	0.4	-0.1
1	0.7	-0.2	0.9	0.4	0.3	-0.5

Tabel 4.12 Bobot serta Bias w

w	
z_1	0.1
z_2	0.5
z_3	0.6
z_4	-0.8
z_5	0.8
z_6	-0.6
1	-0.9

Digunakan data normalisasi seperti pada Tabel 4.2 sebagai data training. Langkah selanjutnya menghitung nilai z_{in} untuk data pertama. Nilai z_{in} yang dihitung adalah sebanyak unit bobot v yang ada yaitu 6. Nilai z_{in} dihitung dengan Persamaan 2.10. Nilai z_{in} dilihat dengan Tabel 4.13.

Tabel 4.13 Nilai z_{in}

z_{in1}	1.31
z_{in2}	1.036666667

z_in3	-0.183333333
z_in4	0.406666667
z_in5	-2.17
z_in6	-0.983333333

Selanjutnya adalah menghitung nilai z dengan cara memasukkan nilai z_{in} kedalam fungsi aktivasi yaitu *binary* sigmoid menggunakan Persamaan 2.11. Nilai z ditunjukkan dengan Tabel 4.14.

Tabel 4.14 Nilai z

z1	0.635680921
z2	0.686398079
z3	0.454294612
z4	0.600288339
z5	0.102477033
z6	0.272230878

Selanjutnya menghitung nilai y_{in} . Terdapat satu unit y_{in} dikarenakan hanya terdapat satu *output*. Nilai y_{in} dihitung dengan Persamaan 2.12. Hasil y_{in} yang didapat yaitu:

$$y_{in} = -0.741156326$$

Selanjutnya adalah menghitung nilai y menggunakan fungsi aktivasi dengan Persamaan 2.13. Hasil y yang didapat adalah:

$$y = 0.322751339$$

4.2.6 Proses *Backpropagation Error*

Tahap awal proses *Backpropagation Error* adalah menghitung nilai δ_k dengan Persamaan 2.14. Target yang digunakan adalah target dari data pertama serta *outputnya* adalah nilai y. Nilai δ_k yang didapat yaitu:

$$\delta_k = (t_k - y_k)y_k(1 - y_k)$$

$$\delta_k = (0.03333 - 0.32275)0.32275(1 - 0.32275)$$

$$\delta_k = -0.06326183$$

Langkah selanjutnya adalah menghitung nilai delta perubahan bobot dan bias w (Δw) yang dihitung dengan Persamaan 2.15 dan 2.16. Nilai Δw ini nantinya digunakan untuk proses pembaruan bobot dan bias w. *Learning rate* yang digunakan adalah 0,1. Hasil perhitungan Δw ditunjukkan pada Tabel 4.15.

$$\Delta W_{jk} = \alpha \delta_k z_j$$

$$\Delta W_{11} = 0,1 \times (-0.06326183) \times 0.787513156$$

$$\Delta W_{11} = -0.004981952$$

$$\Delta W_{0j} = \alpha \delta_k$$

$$\Delta W_{01} = 0,1 \times (-0.06326183)$$

$$\Delta W_{01} = -0.006326183$$

Tabel 4.15 Nilai Δw

Δw_{11}	-0.004981952
Δw_{12}	-0.004670028
Δw_{13}	-0.002873951
Δw_{14}	-0.003797534
Δw_{15}	-0.000648288
Δw_{16}	-0.001722182
Δw_{01}	-0.006326183

Langkah selanjutnya adalah menghitung nilai δ_{in} seperti pada Persamaan 2.17. Hasil δ_{in} dapat dilihat pada Tabel 4.16.

$$\delta_{inj} = \sum_{k=1}^m \delta_i w_{ij}$$

$$\delta_{in1} = -0.06326183 \times 0.1$$

$$\delta_{in1} = -0.006326183$$

Tabel 4.16 Hasil δ_{in}

δ_{in1}	-0.006326183
δ_{in2}	-0.031630915
δ_{in3}	-0.037957098
δ_{in4}	0.050609464
δ_{in5}	-0.050609464
δ_{in6}	0.037957098

Langkah selanjutnya adalah menghitung nilai δ menggunakan Persamaan 2.18. Hasil perhitungan ditunjukkan dengan Tabel 4.17.

$$\delta_j = \delta_{inj} z_j (1 - z_j)$$

$$\delta_1 = (-0.006326183) \times 0.787513156 (1 - 0.787513156)$$

$$\delta_1 = -0.001058599$$

Tabel 4.17 Hasil δ

δ_1	-0.001058599
δ_2	-0.006112919
δ_3	-0.009409983
δ_4	0.012143349
δ_5	-0.00465483
δ_6	0.007520107

Setelah mendapatkan nilai δ selanjutnya adalah menghitung nilai perubahan bobot dan bias v (Δv) dengan Persamaan 2.19 dan 2.20. Hasil perhitungan ditunjukkan dengan Tabel 4.18.

$$\Delta v_{jk} = \alpha \delta_j x_j$$

$$\Delta v_{11} = 0.1 \times (-0.001058599) \times 1$$

$$\Delta v_{11} = -\mathbf{0.00010586}$$

$$\Delta v_{0j} = \alpha \delta_j$$

$$\Delta v_{0j} = 0.1 \times (-0.001058599)$$

$$\Delta v_{0j} = -\mathbf{0.00010586}$$

Tabel 4.18 Hasil Δv

Δv	z1	z2	z3	z4	z5	z6
x1	-0.00010586	-0.000611292	-0.000940998	0.001214335	-0.000465483	0.000752011
x2	-5.99873E-05	-0.000346399	-0.000533232	0.000688123	-0.000263774	0.000426139
x3	-3.1758E-05	-0.000183388	-0.000282299	0.0003643	-0.000139645	0.000225603
x4	-1.41147E-05	-8.15056E-05	-0.000125466	0.000161911	-6.20644E-05	0.000100268
x5	0	0	0	0	0	0
x6	0	0	0	0	0	0
1	-0.00010586	-0.000611292	-0.000940998	0.001214335	-0.000465483	0.000752011

4.2.7 Proses *Update* Bobot dan Bias

Langkah pertama dalam tahap pembaruan (*update*) bobot dan bias adalah memperbarui bobot dan bias v menggunakan Persamaan 2.21 dan 2.22. Hasil perhitungan ditunjukkan dengan Tabel 4.19.

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$$

$$v_{11}(\text{baru}) = v_{11}(\text{lama}) + \Delta v_{11}$$

$$v_{11}(\text{baru}) = 0.7 + -0.00010586$$

$$v_{11}(\text{baru}) = \mathbf{0.69989414}$$

$$v_{0j}(\text{baru}) = v_{0j}(\text{lama}) + \Delta v_{0j}$$

$$v_{01}(\text{baru}) = v_{01}(\text{lama}) + \Delta v_{01}$$

$$v_{01}(\text{baru}) = 0.4 + -0.00010586$$

$$v_{01}(\text{baru}) = \mathbf{0.39989414}$$

Tabel 4.19 Hasil Pembaruan Bobot serta Bias v

v baru	z1	z2	z3	z4	z5	z6
x1	0.699894	0.699388	-0.60094	0.101214	-0.800465	-0.599247
x2	-0.200059	-0.600346	0.399466	0.800688	-0.900263	-0.599573
x3	0.899968	-0.500183	-0.300282	0.800364	-0.800139	-0.29977
x4	0.399985	0.199918	0.599874	0.100161	-0.900062	-0.399899
x5	0.3	0.7	-0.7	0.2	0.8	0.4
x6	-0.5	0.2	0.7	0.3	0.7	-0.1

1	0.399894	0.799388	0.199059	-0.398785	-0.500465	0.100752
---	-----------------	----------	----------	-----------	-----------	----------

Langkah selanjutnya adalah menghitung pembaruan bobot dan bias w dengan Persamaan 2.23 dan 2.24. Hasil perhitungan ditunjukkan dengan Tabel 4.20.

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$$

$$w_{11}(\text{baru}) = w_{11}(\text{lama}) + \Delta w_{11}$$

$$w_{11}(\text{baru}) = 0.1 + -0.004981952$$

$$w_{11}(\text{baru}) = \mathbf{0.095018048}$$

$$w_{0j}(\text{baru}) = w_{0j}(\text{lama}) + \Delta w_{0j}$$

$$w_{0j}(\text{baru}) = (-0.9) + (-0.006326183)$$

$$w_{0j}(\text{baru}) = \mathbf{-0.906326183}$$

Tabel 4.20 Hasil Pembaruan Bobot serta Bias w

W(baru)	y
z1	0.095018048
z2	0.495329972
z3	0.597126049
z4	-0.80379753
z5	0.799351712
z6	-0.60172218
1	-0.90632618

Proses ini diulang sebanyak data latih yang ditentukan yaitu empat data latih.

4.2.8 Proses Pengujian Backpropagation

Tahap *feedforward*, *backpropagation error*, dan *update* bobot dilakukan sampai dengan 2 iterasi. Nilai bobot dan bias dari v dan w ini kemudian akan digunakan untuk prediksi menggunakan data uji yang telah ditentukan yaitu memprediksi jumlah kasus penyakit pada bulan Januari 2016. Data uji ditunjukkan pada Tabel 4.21. Bobot serta bias v dan w pada hasil akhir pelatihan ditunjukkan dengan Tabel 4.22 dan 4.23.

Tabel 4.21 Data Uji

Input						t
x1	x2	x3	x4	x5	x6	
0	0	0.025641026	0.205128205	0.666666667	0.333333333	?

Tabel 4.22 Bobot serta Bias v

V(baru)	z1	z2	z3	z4	z5	z6
x1	0.699903 6	0.699252 6	-0.601156	0.101434 8	-0.799934	-0.599258

x2	-0.200073	-0.600501	0.399225 2	0.800974 9	-0.900113	-0.599458
x3	0.899926 6	-0.500415	-0.300637	0.800823 7	-0.800342	-0.299482
x4	0.399971 8	0.199837 0	0.599749 4	0.100322 1	-0.900121	-0.399801
x5	0.300020 5	0.700086 2	-0.699868	0.199820 0	0.8001653	0.399865 5
x6	-0.499853	0.200623 9	0.700950 2	0.298708 8	0.7011819	-0.100965
1	0.400246 2	0.800712 0	0.201062 8	-0.401577	- 0.4971303	0.098479

Tabel 4.23 Bobot serta Bias w

W(baru)	
z1	0.102009503
z2	0.505426802
z3	0.605752901
z4	-0.79858641
z5	0.806171988
z6	-0.59398433
1	-0.89256242

Proses pengujian dilakukan dengan menjalankan proses *feedforward* menggunakan bobot dan bias pada Tabel 4.22 dan 4.23. Dari proses *feedforward* maka didapat nilai y :

$$y = \mathbf{0.408894378}$$

Kemudian akan dilakukan proses denormalisasi data untuk mendapatkan data aktual dengan Persamaan 4.4:

$$x = x'(x_{max} - x_{min}) + x_{min}$$

$$x = 0.408894378 \times (33 - 7) + 0$$

$$x = \mathbf{18}$$

Hasil prediksi menunjukkan jumlah kasus penyakit pada bulan juli tahun 2016 sebanyak 11, sementara jumlahnya sebesar 16. Setelah mendapat hasil prediksi langkah selanjutnya adalah menghitung nilai *error* menggunakan MSE seperti pada Persamaan 2.25. Hasil perhitungan adalah sebagai berikut:

$$MSE = \frac{1}{n} \sum_{i=1}^n (t_k - y_k)^2$$

$$MSE = \frac{(33-18)^2}{1}$$

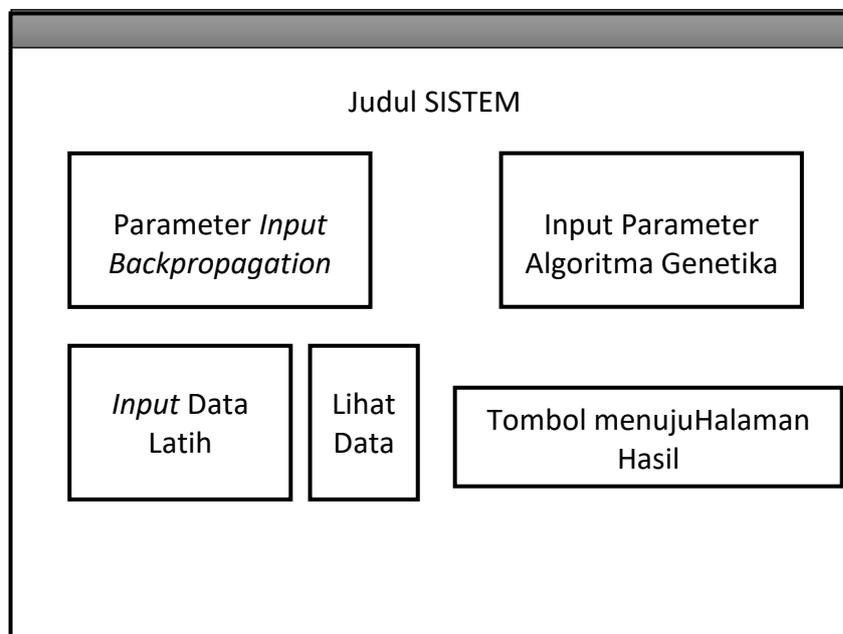
$$MSE = \mathbf{236.198}$$

4.3 Perancangan Antarmuka

Perancangan ini bertujuan untuk memberi gambaran antarmuka yang nantinya akan diimplementasikan. Antarmuka merupakan perantara yang menjembatani sistem dan pengguna dengan cara menerima masukan dari pengguna serta menampilkan keluaran dari sistem. Rancangan antarmuka ini terbagi dari halaman utama, halaman data, serta halaman hasil.

4.3.1 Perancangan Antarmuka Halaman Utama

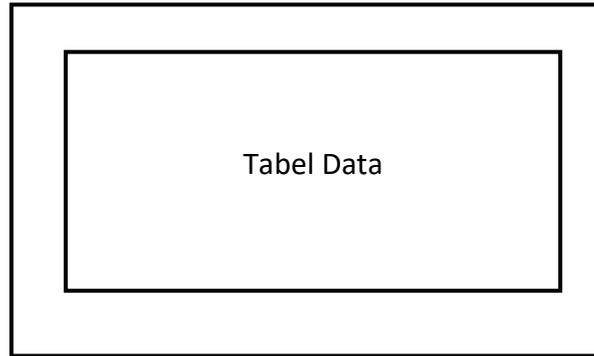
Perancangan halaman utama terdiri dari nama sistem, parameter algoritma genetika, parameter *backpropagation*, data latih serta data uji yang digunakan. Halaman ini berfungsi untuk menghubungkan pengguna dengan sistem melalui masukan dari parameter serta data yang akan digunakan. Rancangan antarmuka halaman utama ditunjukkan dengan Gambar 4.8:



Gambar 4.9 Rancangan Antarmuka Halaman Utama

4.3.2 Perancangan Antarmuka Halaman Data

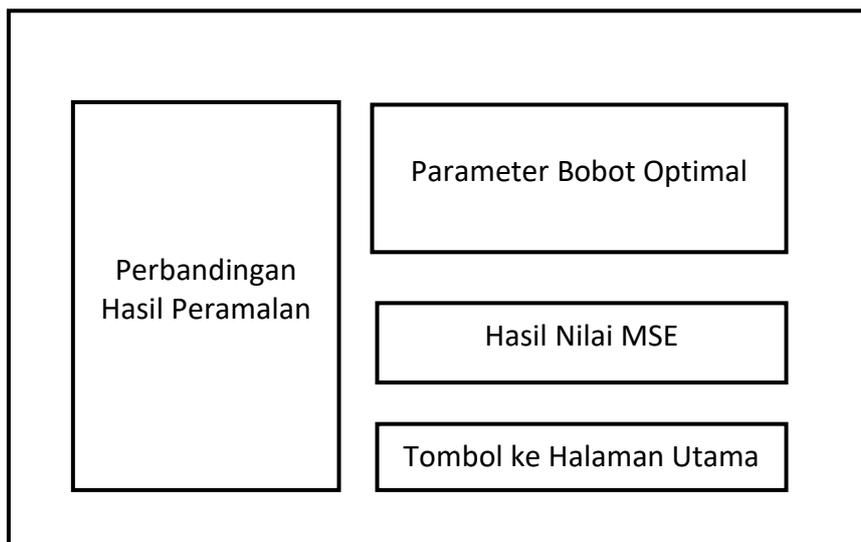
Perancangan halaman data terdiri tabel yang berisi data yang akan digunakan. Halaman ini bertujuan untuk menampilkan data yang digunakan pada peramalan jumlah kasus penyakit kepada pengguna. Rancangan antarmuka halaman data ditunjukkan dengan Gambar :



Gambar 4.10 Rancangan Antarmuka Halaman Data

4.3.3 Perancangan Antarmuka Halaman Hasil

Halaman hasil terdiri dari tabel hasil, nilai parameter optimal, nilai MSE, tombol ke halaman utama. Halaman ini berfungsi untuk menunjukkan hasil peramalan dari data yang dimasukkan oleh pengguna berupa. Hasil peramalan merupakan keluaran dari peramalan dengan data uji, nilai parameter optimal serta nilai MSE. Rancangan halaman hasil ditunjukkan dengan Gambar 4.11:



Gambar 4.11 Rancangan Antarmuka Halaman Hasil