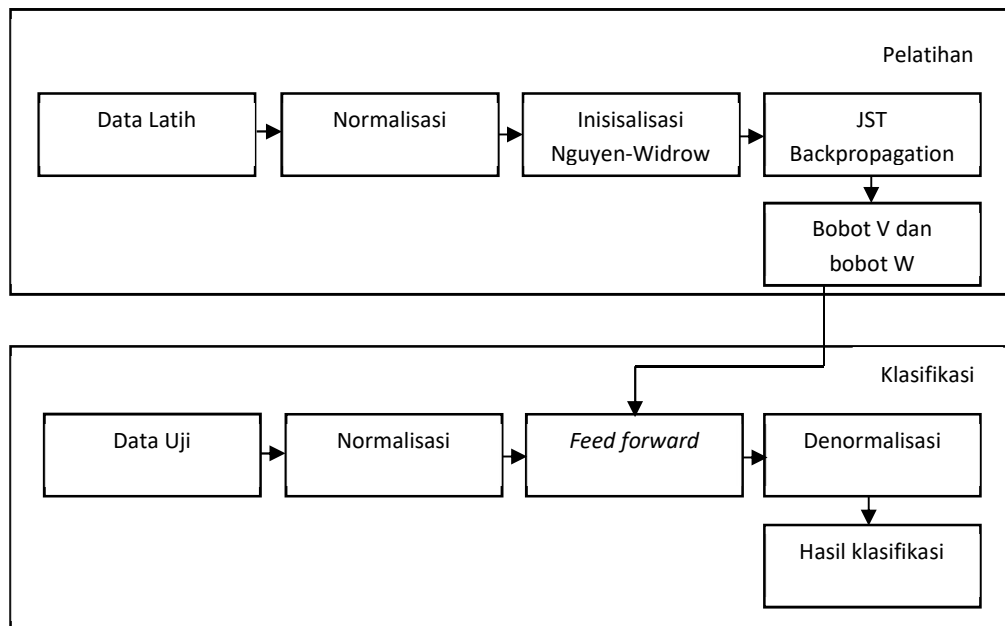


## BAB 4 PERANCANGAN

### 4.1 Perancangan Proses

Perancangan proses klasifikasi rumah layak huni menggunakan metode *Backpropagation* dengan inialisasi bobot *nguyen-widrow* secara umum terbagi menjadi dua bagian utama. Perancangan proses secara umum dapat dilihat pada Gambar 4.1.



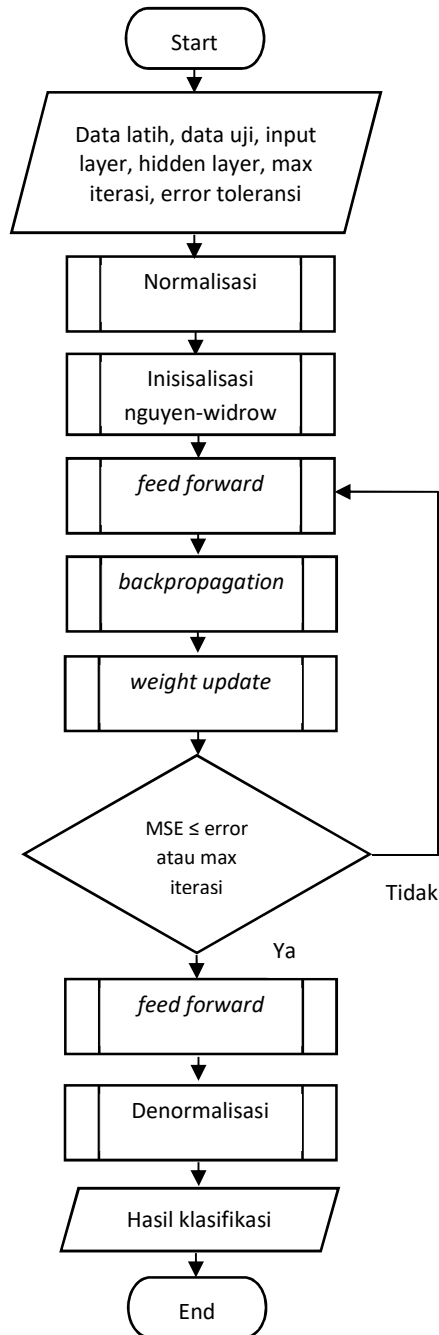
**Gambar 4. 1** Arsitektur Proses Klasifikasi Rumah Layak Huni

Tahapan proses dalam sistem klasifikasi rumah layak huni terbagi menjadi dua bagian yaitu pelatihan Jaringan Syaraf Tiruan dengan inialisasi bobot *nguyen-widrow* dan klasifikasi rumah layak huni. Pelatihan jaringan merupakan proses untuk melatih jaringan dengan menghasilkan bobot yang sesuai dengan data yang dilatih untuk digunakan pada proses klasifikasi. Proses pelatihan jaringan membutuhkan masukan berupa data latih dan inialisasi bobot dengan *nguyen-widrow*. Proses pelatihan akan melakukan *update* bobot hingga bobot mencapai nilai *error* minimum.

Proses klasifikasi merupakan proses dimana dilakukan proses *feed forward* dengan masukan berupa data uji. Selain itu, bobot yang didapat dari proses pelatihan digunakan pada proses klasifikasi dengan *feed forward*. Kemudian akan dilakukan proses denormalisasi yaitu proses merubah rentang nilai menjadi rentang nilai asli. Hasil dari proses denormalisasi merupakan hasil klasifikasi rumah layak huni.

## 4.2 Perancangan Algoritma *Backpropagation*

Perancangan algoritma *Backpropagation* yang digunakan sistem dalam klasifikasi rumah layak huni terdiri dari perancangan proses normalisasi, *feed forward* (propagasi maju), *backpropagation* (proses balik), dan proses denormalisasi. Dalam mempermudah memahami jalannya sistem dibutuhkan diagram alir. Diagram alir merupakan langkah-langkah dari suatu proses secara keseluruhan. Gambar 4.2 menunjukkan diagram alir sistem secara umum.



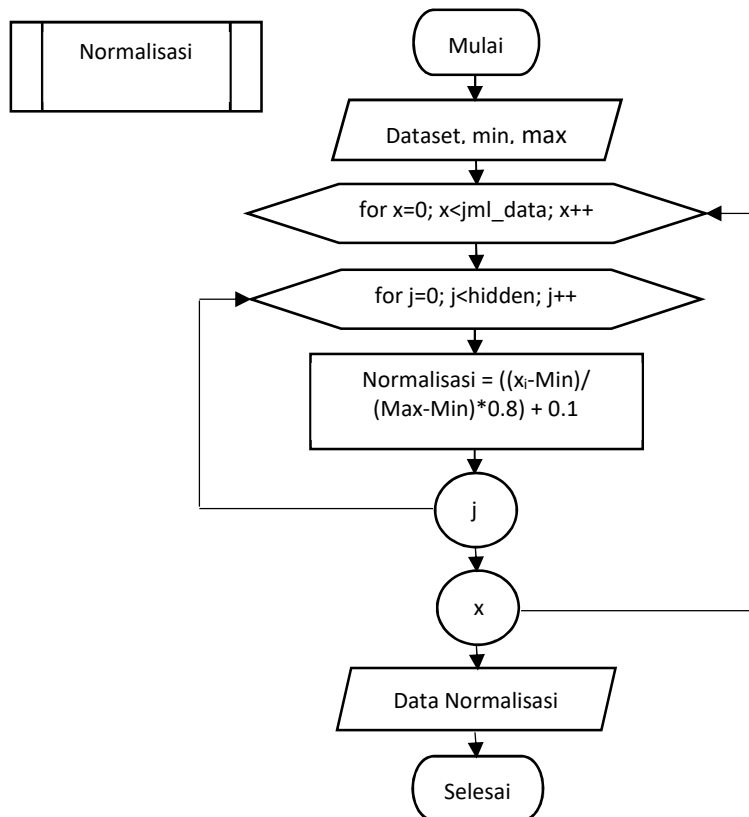
Gambar 4. 2 Diagram Alir Sistem

Tahapan alir sistem secara umum pada Gambar 4.2 adalah sebagai berikut:

1. Menetapkan *input layer*, *hidden layer*, maksimum iterasi, *error* toleransi dan menyiapkan data latih serta data uji yang akan digunakan.
2. Melakukan tahap normalisasi.
3. Inisialisasi bobot dan bias menggunakan *nguyen-widrow*.
4. Melakukan tahap *feed forward*.
5. Melakukan tahap *backpropagation*.
6. Melakukan tahap *weight update*.
7. Jika nilai MSE kurang dari nilai *error* yang ditoleransi maka akan dilanjutkan tahap denormalisasi. Namun, jika MSE lebih besar dari nilai *error* yang ditoleransi dan iterasi kurang dari maksimum iterasi maka akan kembali ke tahap *feed forward*.
8. Melakukan pengujian, dilakukan dengan proses *feed forward*.
9. Melakukan tahap denormalisasi untuk mengembalikan nilai data asli.

#### 4.2.1 Normalisasi

Normalisasi merupakan proses perubahan nilai data ke skala tertentu. Gambar 4.3 menunjukkan diagram alir proses normalisasi.



**Gambar 4. 3 Diagram Alir Fungsi Normalisasi**

Tahap normalisasi pada Gambar 4.3 terdiri dari:

1. Input berupa data rumah layak huni dan rumah tidak layak huni yang akan dinormalisasi , nilai min, dan nilai maksimum.
2. Melakukan perulangan sebanyak jumlah kolom.
3. Melakukan perulangan sebanyak jumlah *input layer*.
4. Menghitung normalisasi.
5. Output berupa data yang telah dinormalisasi.

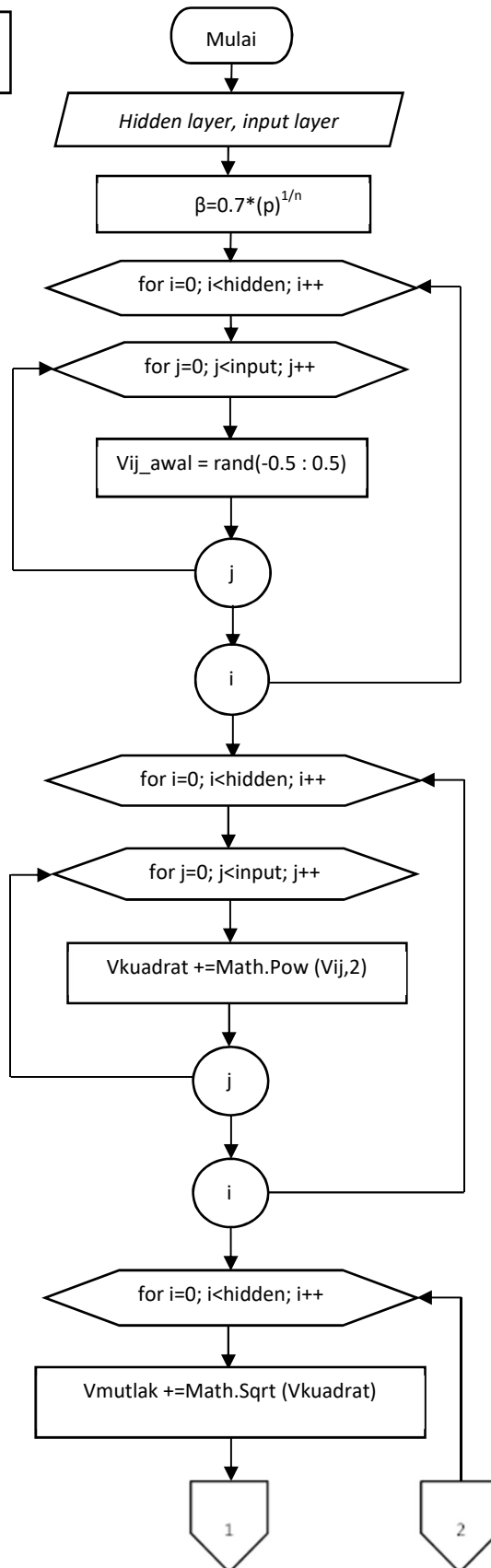
#### **4.2.2 Nguyen Widrow**

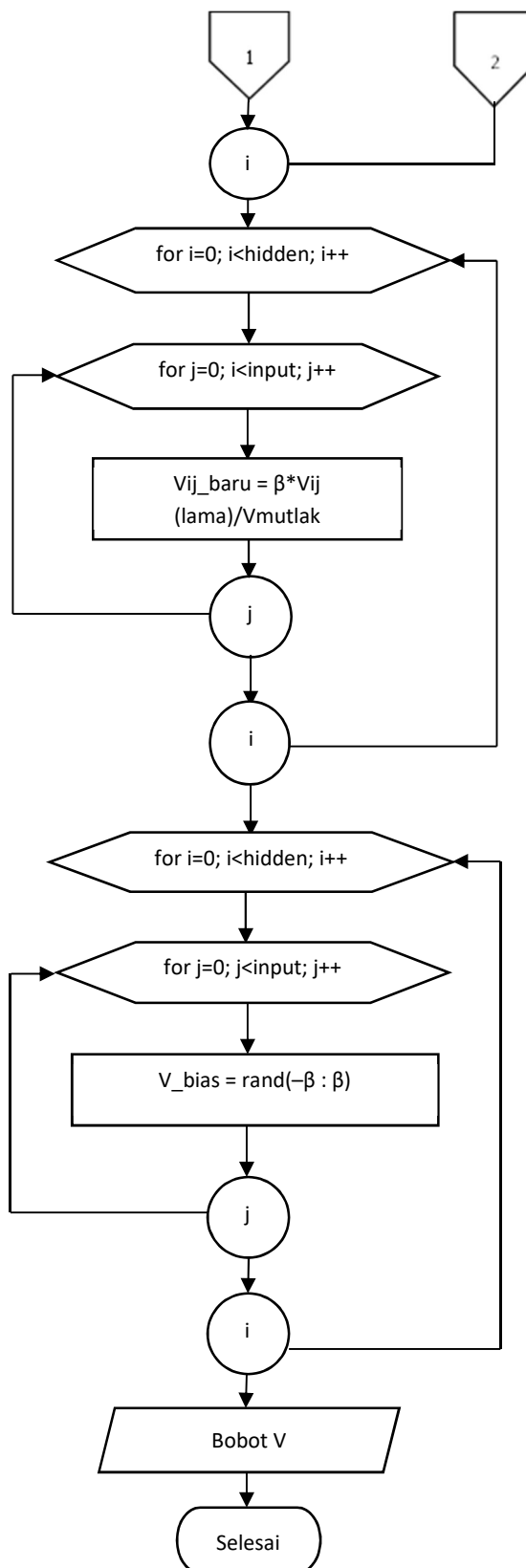
*Nguyen Widrow* merupakan proses inialisasi bobot yang akan digunakan untuk perhitungan algoritma *backpropagation*. Pada proses ini didapatkan bobot antara *input layer* dengan *hidden layer* (V) dan bobot antara *hidden layer* dengan *output layer* (W). Gambar 4.4 menunjukkan diagram alir fungsi *nguyen widrow* .

Proses *nguyen-widrow* yang terdapat pada Gambar 4.4 terdiri dari:

1. *Input* berupa *input layer* dan *hidden layer*.
2. Menghitung faktor skala  $\beta$ .
3. Melakukan perulangan sebanyak *hidden layer*.
4. Melakukan perulangan sebanyak *input layer*.
5. Inialisasi bobot  $V_{ij}$  secara acak  $[-0,5 : 0,5]$ .
6. Melakukan perulangan sebanyak *hidden layer*.
7. Melakukan perulangan sebanyak *input layer*.
8. Menghitung  $V_{kuadrat}$  yaitu penjumlahan bobot yang dikuadrat untuk digunakan pada perhitungan  $V_{mutlak}$  selanjutnya.
9. Melakukan perulangan sebanyak *hidden layer*.
10. Menghitung  $V_{mutlak}$  dengan akar dari  $V_{kuadrat}$ .
11. Melakukan perulangan sebanyak *hidden layer*.
12. Melakukan perulangan sebanyak *input layer*.
13. Melakukan update bobot  $V_{ij}$  dengan mengalikan faktor skala dengan bobot lama kemudian dibagi dengan  $V_{mutlak}$ .
14. Melakukan perulangan sebanyak *hidden layer*.
15. Melakukan perulangan sebanyak *input layer*.
16. Menentukan bias antara unit ke unit tersembunyi  $V_{0j} = \text{bilangan acak } [-\beta : \beta]$ .
17. *Output* berupa bobot V baru.

Inisialisasi bobot  
nguyen-widrow





**Gambar 4. 4 Diagram Alir Fungsi Nguyen Widrow**

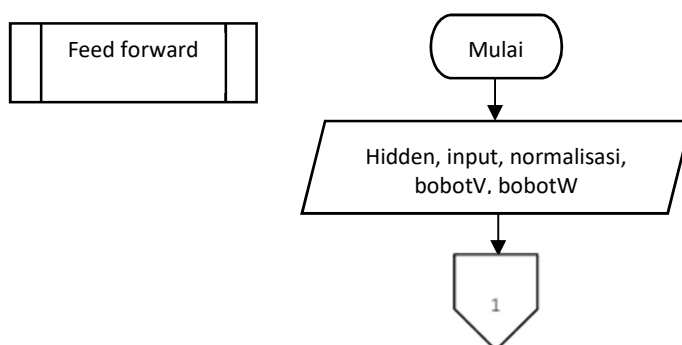
### 4.2.3 Feed Forward

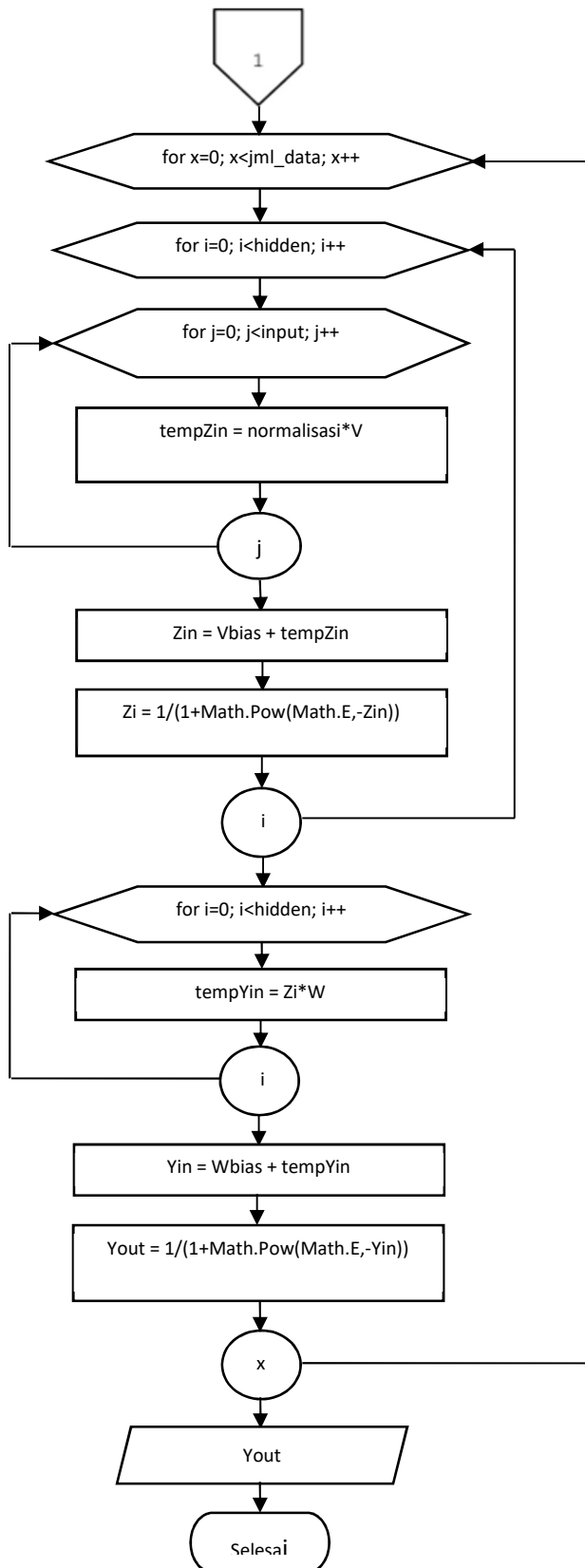
*Feed forward* merupakan proses awal yang dilakukan dalam algoritma *Backpropagation*. Pada proses ini *input layer* dihubungkan dengan *hidden layer* dengan menggunakan fungsi aktivasi. Setelah unit *input* diterima maka akan diteruskan ke *hidden layer*. Gambar 4.5 menunjukkan diagram alir fungsi *feed forward*.

Tahap *feed forward* pada Gambar 4.5 terdiri dari:

1. *Input* berupa *input layer*, *hidden layer*, data yang telah dinormalisasi, serta bobot  $V$  dan bobot  $W$  yang didapat dari inisialisasi *nguyen-widrow*.
2. Melakukan perulangan sebanyak jumlah data.
3. Melakukan perulangan sebanyak *hidden layer*.
4. Melakukan perulangan sebanyak *input layer*.
5. Menghitung  $Z_{in}$  sementara dimana penjumlahan perkalian normalisasi dengan bobot  $V$ .
6. Menghitung neuron yang masuk ke *hidden layer* dari *input layer* ( $Z_{in}$ ).
7. Menghitung neuron yang keluar dari *hidden layer* menuju *output layer* ( $Z_i$ ).
8. Melakukan perulangan sebanyak *hidden layer*.
9. Menghitung bobot  $W$  sementara dimana penjumlahan perkalian  $Z_i$  dengan bobot  $W$ .
10. Menghitung neuron yang masuk ke *output layer* dari *hidden layer* ( $Y_{in}$ ).
11. Menghitung neuron yang masuk ke *output layer* dari *hidden layer* ( $Y_{out}$ ).
12. *Output* berupa  $Y_{out}$ .

Berikut adalah diagram alir fungsi *feed forward*:





**Gambar 4. 5 Diagram Alir Fungsi Feed Forward**



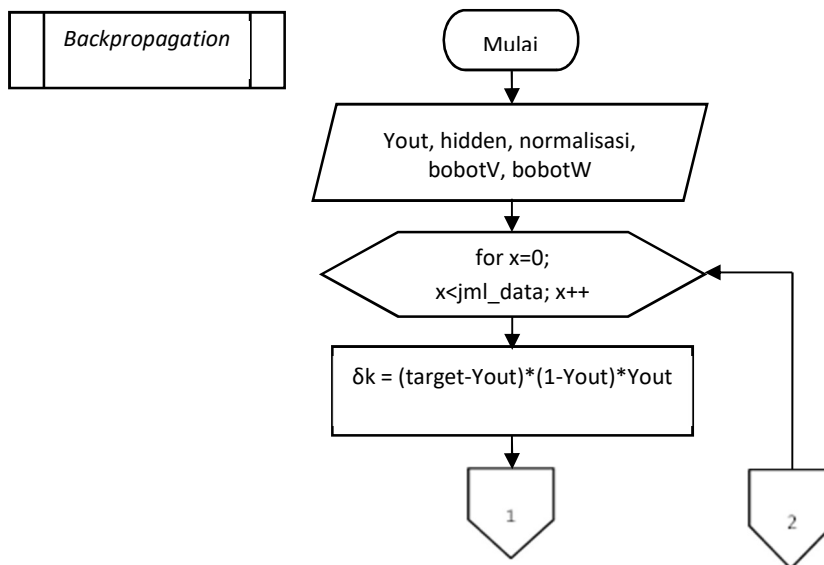
#### 4.2.4 Backpropagation

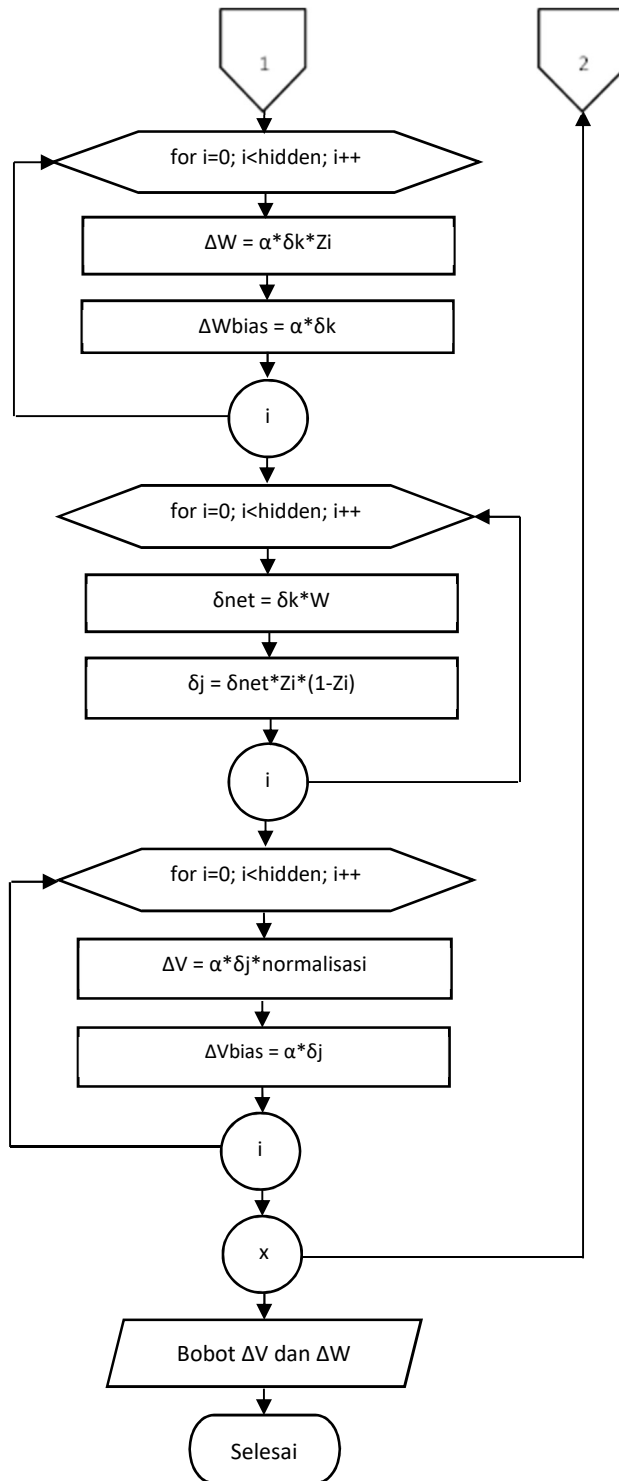
*Backpropagation* merupakan tahap selanjutnya setelah proses *feed forward*, dimana hasil output *hidden layer* dilanjutkan untuk menghitung *output layer*. Dalam proses ini terdapat koreksi kesalahan dan perbaikan bobot dan bias. Gambar 4.6 menunjukkan diagram alir fungsi *backpropagation*.

Tahap *backpropagation* pada Gambar 4.6 terdiri dari:

1. Inputan awal berupa  $Y_{out}$ , *input layer*, *hidden layer*, normalisasi, bobot  $V$  dan bobot  $W$ .
2. Melakukan perulangan sebanyak jumlah data.
3. Menghitung nilai *error* antara *output layer* dan *hidden layer* ( $\delta_k$ ).
4. Melakukan perulangan sebanyak *hidden layer*.
5. Menghitung koreksi bobot antara *hidden layer* dan *output layer* ( $\Delta W$ ).
6. Menghitung koreksi bias ( $\Delta W$  bias).
7. Melakukan perulangan sebanyak *hidden layer*.
8. Menghitung nilai galat antara *hidden layer* dan *input layer* ( $\delta_{net}$ ).
9. Menghitung faktor kesalahan (galat) di *hidden layer* ( $\delta_j$ ).
10. Melakukan perulangan sebanyak *hidden layer*.
11. Menghitung koreksi bobot baru antara *hidden layer* dan *input layer* ( $\Delta V$ ).
12. Menghitung koreksi bias ( $\Delta V$  bias).
13. Output berupa nilai bobot  $\Delta V$  dan  $\Delta W$ .

Berikut adalah diagram alir dari tahap *backpropagation*:

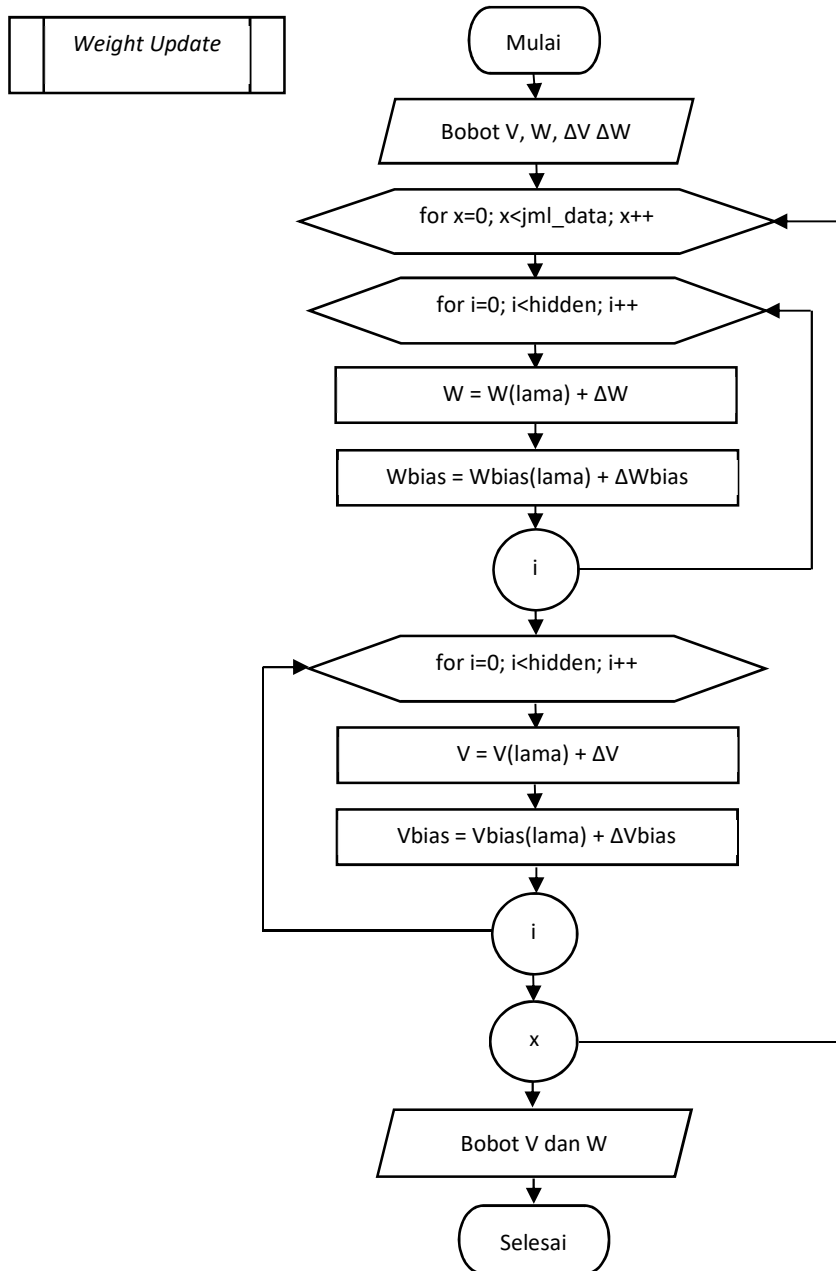




**Gambar 4. 6 Diagram Alir Fungsi *Backpropagation***

#### **4.2.5 *Weight Update***

*Weight Update* merupakan tahap selanjutnya setelah proses *Backpropagation*, dimana akan dilakukan perubahan bobot baru terhadap hasil yang dikeluarkan pada proses sebelumnya. Gambar 4.7 menunjukkan diagram alir fungsi *Weight Update*.



**Gambar 4. 7 Diagram Alir Fungsi *Weight Update***

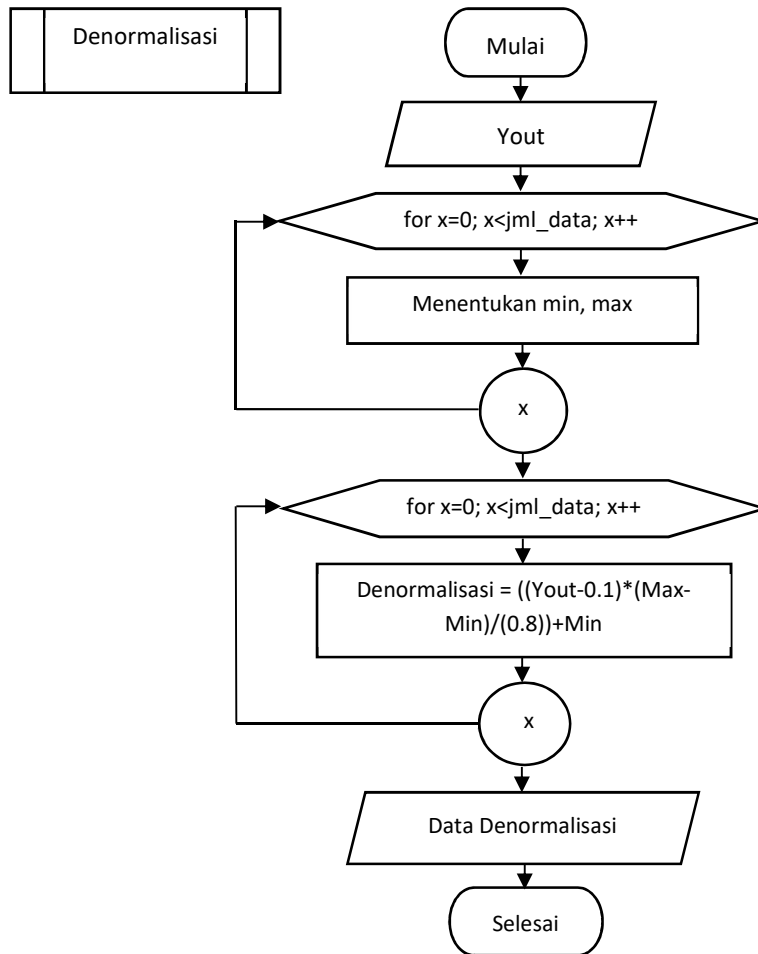
Tahap *backpropagation* pada Gambar 4.7 terdiri dari:

1. Inputan awal berupa bobot V, bobot W, bobot  $\Delta V$ , dan bobot  $\Delta W$ .
2. Melakukan perulangan sebanyak jumlah data.
3. Melakukan perulangan sebanyak *hidden layer*.
4. Menghitung perubahan bobot nilai W yang baru.
5. Menghitung perubahan bobot nilai W bias.
6. Melakukan perulangan sebanyak *hidden layer*.

7. Menghitung perubahan bobot nilai V yang baru.
8. Menghitung perubahan bobot nilai V bias.
9. Output berupa nilai bobot V dan W.

#### 4.2.6 Denormalisasi

Denormalisasi merupakan proses merubah rentang nilai sebelumnya yang memiliki rentang nilai tertentu menjadi rentang nilai asli. Gambar 4.8 menunjukkan diagram alir fungsi denormalisasi.



**Gambar 4. 8 Diagram Alir Fungsi Denormalisasi**

Tahap denormalisasi pada Gambar 4.8 terdiri dari:

1. Input berupa data hasil perhitungan *backpropagation* dengan inisialisasi bobot *nguyen-widrow* yaitu Yout.
2. Melakukan perulangan sebanyak jumlah data.
3. Menentukan nilai maksimum dan minimum.
4. Melakukan perulangan sebanyak jumlah data.
5. Menghitung denormalisasi.

### 4.3 Manajemen Data

Pada subbab manajemen data akan dijelaskan tentang aturan-aturan dan kriteria-kriteria yang digunakan dalam perhitungan dan implementasi metode *Backpropagation* untuk klasifikasi rumah layak huni.

Penentuan kriteria dilakukan berdasarkan sumber dari data Desa Kidal Kecamatan Tumpang Kabupaten Malang. Dalam penelitian ini telah ditentukan 15 kriteria yang digunakan dalam penentuan rumah layak huni. Berikut adalah kriteria-kriteria yang telah ditetapkan:

1. Penghasilan perbulan
2. Pekerjaan
3. Jumlah anggota keluarga
4. Jumlah KK dalam rumah
5. Status kepemilikan rumah
6. Kondisi rumah
7. Luas bangunan
8. Luas tanah
9. Jenis lantai rumah
10. Bahan dinding rumah
11. Kondisi dinding
12. Bahan atap rumah
13. Kamar MCK
14. Air bersih
15. Listrik

Setiap kriteria memiliki masing-masing parameter, dimana parameter tersebut digunakan untuk memberikan nilai pada data yang digunakan dalam perhitungan dengan metode *Backpropagation*.

1. Penghasilan perbulan

Parameter ukuran penghasilan perbulan meliputi nilai 4 untuk penghasilan <300, nilai 3 penghasilan 300-600, nilai 2 penghasilan 601-900, dan nilai 1 penghasilan >900 yang ditunjukkan pada Tabel 4.1.

**Tabel 4. 1 Parameter Ukuran Penghasilan Perbulan**

Parameter Ukuran	Nilai
<300	4
300 - 600	3
601 - 900	2
>900	1

2. Pekerjaan

Parameter ukuran pekerjaan meliputi nilai 6 untuk tidak bekerja, nilai 5 untuk buruh tani, nilai 4 untuk petani, nilai 3 untuk buruh, nilai 2 untuk pegawai, nilai 1 untuk pedagang yang ditunjukkan pada Tabel 4.2.

**Tabel 4. 2 Parameter Ukuran Pekerjaan**

Parameter Ukuran	Nilai
Tidak Bekerja	6
Buruh Tani	5
Petani	4
Buruh	3
Pegawai	2
Pedagang	1

3. Jumlah anggota keluarga

Parameter ukuran jumlah anggota keluarga meliputi nilai 3 untuk anggota >5, nilai 2 untuk anggota  $3 \leq 5$ , dan nilai 1 untuk anggota <3 yang ditunjukkan pada Tabel 4.3.

**Tabel 4. 3 Parameter Ukuran Jumlah Anggota Keluarga**

Parameter Ukuran	Nilai
> 5	3
$3 \leq 5$	2
< 3	1

4. Jumlah KK dalam rumah

Parameter ukuran jumlah KK dalam rumah meliputi nilai 5 untuk jumlah 1, nilai 3 untuk jumlah 3, nilai 1 untuk jumlah <2 yang ditunjukkan pada Tabel 4. 4.

**Tabel 4. 4 Parameter Ukuran Jumlah KK dalam Rumah**

Parameter Ukuran	Nilai
1	5
2	3
< 2	1

5. Status kepemilikan rumah

Parameter ukuran status kepemilikan rumah meliputi nilai 3 untuk sertifikat hak milik dan nilai 1 untuk sertifikat milik anak yang ditunjukkan pada Tabel 4.5.

**Tabel 4. 5 Parameter Ukuran Status Kepemilikan Rumah**

Parameter Ukuran	Nilai
Sertifikat Hak Milik	3
Sertifikat Milik Anak	1

6. Kondisi rumah

Parameter ukuran kondisi rumah meliputi nilai 5 untuk rusak, nilai 3 untuk tidak permanen, nilai 1 untuk permanen yang ditunjukkan pada Tabel 4.6.

**Tabel 4. 6 Parameter Ukuran Kondisi Rumah**

Parameter Ukuran	Nilai
Rusak	5
Tidak Permanen	3
Permanen	1

7. Luas bangunan

Parameter ukuran luas bangunan meliputi nilai 7 untuk luas >25, nilai 5 untuk luas 25-55, nilai 3 untuk luas 56-85, nilai 1 untuk luas >85 yang ditunjukkan pada Tabel 4.7.

**Tabel 4. 7 Parameter Ukuran Luas Bangunan**

Parameter Ukuran	Nilai
> 25	7
25 – 55	5
56 – 85	3
> 85	1

8. Luas tanah

Parameter ukuran luas tanah meliputi nilai 7 untuk luas  $\geq 30$ , nilai 5 untuk luas 31-60, nilai 3 untuk luas 61-90, nilai 1 untuk luas >90 yang ditunjukkan pada Tabel 4.8.

**Tabel 4. 8 Parameter Ukuran Luas Tanah**

Parameter Ukuran	Nilai
$\geq 30$	7
31 – 60	5
61 – 90	3
> 90	1

9. Jenis lantai rumah

Parameter ukuran jenis lantai rumah meliputi nilai 6 untuk tanah rusak, nilai 5 untuk tanah baik, nilai 4 untuk plester rusak, nilai 3 untuk plester baik, nilai 2 untuk keramik rusak, nilai 1 untuk keramik baik yang ditunjukkan pada Tabel 4.9.

**Tabel 4. 9 Parameter Ukuran Jenis Lantai Rumah**

Parameter Ukuran	Nilai
Tanah (rusak)	6
Tanah (baik)	5
Plester (rusak)	4
Plester (baik)	3
Keramik (rusak)	2
Keramik (baik)	1

10. Bahan dinding rumah

Parameter ukuran bahan dinding rumah meliputi nilai 5 untuk anyaman bambu, nilai 3 untuk papan, nilai 1 untuk tembok yang ditunjukkan pada Tabel 4.10.

**Tabel 4. 10 Parameter Ukuran Bahan Dinding Rumah**

Parameter Ukuran	Nilai
Anyaman Bambu	5
Papan	3
Tembok	1

11. Kondisi dinding

Parameter ukuran kondisi dinding meliputi nilai 5 untuk rusak, nilai 3 untuk rusak sedang, nilai 1 untuk baik yang ditunjukkan pada Tabel 4.11.

**Tabel 4. 11 Parameter Ukuran Kondisi Dinding**

Parameter Ukuran	Nilai
Rusak	5
Rusak Sedang	3
Baik	1

12. Bahan atap rumah

Parameter ukuran bahan atap rumah meliputi nilai 5 untuk genteng rusak, nilai 3 untuk genteng rusak sedang, nilai 1 untuk genteng baik yang ditunjukkan pada Tabel 4.12.

**Tabel 4. 12 Parameter Ukuran Bahan Atap Rumah**

Parameter Ukuran	Nilai
Genteng (rusak)	5
Genteng (rusak sedang)	3
Genteng (baik)	1



### 13. Kamar MCK

Parameter ukuran kamar mck meliputi nilai 5 untuk tidak ada, dan nilai 1 untuk ada yang ditunjukkan pada Tabel 4.13.

**Tabel 4. 13 Parameter Ukuran Kamar MCK**

Parameter Ukuran	Nilai
Tidak Ada	5
Ada	1

### 14. Air bersih

Parameter ukuran air bersih meliputi nilai 5 untuk tidak ada dan nilai 1 untuk ada yang ditunjukkan pada Tabel 4.14.

**Tabel 4. 14 Parameter Ukuran Air Bersih**

Parameter Ukuran	Nilai
Tidak Ada	5
Ada	1

### 15. Listrik

Parameter ukuran listrik meliputi nilai 5 untuk tidak ada, nilai 3 untuk ada (nyalur), nilai 1 untuk ada (milik sendiri) yang ditunjukkan pada Tabel 4.15.

**Tabel 4. 15 Parameter Ukuran Listrik**

Parameter Ukuran	Nilai
Tidak Ada	5
Ada (nyalur)	3
Ada (milik sendiri)	1

## 4.4 Perhitungan Manual

Perhitungan manual merupakan contoh perhitungan dari perancangan sistem yang dibuat. Perhitungan manual berfungsi untuk mengetahui apakah perhitungan yang dilakukan benar dan sesuai algoritma yang ada di perancangan atau tidak.

### Perhitungan Data Latih

Data yang digunakan merupakan data rumah layak huni dan rumah tidak layak huni. Tabel 4.16 merupakan data yang akan digunakan.

**Tabel 4. 16 Data Latih**

Data	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	Y
1	2	4	2	5	1	1	5	1	1	1	1	1	1	1	1	2
2	1	3	2	5	3	1	5	5	1	1	1	1	1	1	1	2
3	1	1	2	5	3	1	1	1	1	1	1	1	1	1	1	2
4	1	1	2	3	3	1	1	1	1	1	1	1	1	1	1	2
5	3	4	1	5	3	3	5	5	6	5	3	3	5	5	3	1
6	4	6	1	5	3	3	3	1	6	5	3	3	5	5	3	1
7	2	3	2	5	3	3	5	1	6	5	3	3	5	5	3	1
8	3	4	1	5	3	3	3	1	6	5	5	5	5	5	3	1

Normalisasi Data:

$$X1 = \frac{2 - 1}{4 - 1} * (0,9 - 0,1) + 0,1 = 0,36$$

Normalisasi dilakukan terhadap seluruh data latih pada Tabel 4.16, sehingga didapatkan hasil normalisasi data pada Tabel 4.17 berikut.

**Tabel 4. 17 Data latih yang telah dinormalisasi**

Data	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	Y
1	0.36	0.58	0.5	0.9	0.1	0.1	0.63	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2	0.1	0.42	0.5	0.9	0.9	0.1	0.63	0.63	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
3	0.1	0.1	0.5	0.9	0.9	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
4	0.1	0.1	0.5	0.5	0.9	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
5	0.63	0.58	0.1	0.9	0.9	0.5	0.63	0.63	0.9	0.9	0.5	0.5	0.9	0.9	0.5	0.9
6	0.9	0.9	0.1	0.9	0.9	0.5	0.36	0.1	0.9	0.9	0.5	0.5	0.9	0.9	0.5	0.9
7	0.36	0.42	0.5	0.9	0.9	0.5	0.63	0.1	0.9	0.9	0.5	0.5	0.9	0.9	0.5	0.9
8	0.63	0.58	0.1	0.9	0.9	0.5	0.36	0.1	0.9	0.9	0.9	0.9	0.9	0.9	0.5	0.9

Setelah dinormalisasi, kemudian ditentukan parameter awal yang akan digunakan dalam pelatihan, yaitu:

- *Learning rate* ( $\alpha$ ) = 0,2
- *Epsilon* ( $\epsilon$ ) = 0,01
- Eksponen = 2.71828

Berikut merupakan tahapan perhitungan manual menggunakan *Backpropagation*:

**Iterasi 1 Data 1**

1. Menentukan bobot awal dengan algoritma *Nguyen Widrow*

- Bobot ditentukan acak dengan range [-0,5 : 0,5] ditunjukkan pada tabel 4.18 dan tabel 4.19.

**Tabel 4. 18 Bobot antara *input* layer dengan *hidden* layer (*Vij*)**

Vij	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
Z1	-0.45	0.48	-0.23	0.38	0.25	-0.42	0.31	0.41	0.5	0.33	-0.28	0.26	0.38	0.5	0.29
Z2	0.27	-0.5	0.15	0.45	-0.4	0.28	-0.23	-0.22	0.24	0.42	-0.37	0.46	-0.24	-0.37	0.43
Z3	0.22	-0.26	0.33	0.52	0.35	0.37	-0.39	0.48	-0.45	0.34	-0.42	0.34	0.49	-0.41	0.25

**Tabel 4. 19 Bobot antara *hidden* layer dengan *output* layer (*Wkj*)**

Wkj	Y
Bias	0.5
Z1	0.36
Z2	-0.27
Z3	0.4

- Menghitung nilai  $||V_{ij}||$  dengan bobot  $V_{ij}$  yang telah ditentukan di awal dengan persamaan 2.8 ditunjukkan pada tabel 4.20

$$\begin{aligned}
 |V_{i1}| &= \sqrt{(-0,45)^2 + 0,48^2 + (-0,23)^2 + 0,38^2 + 0,25^2 + (-0,42)^2} \\
 &\quad \sqrt{+0,31^2 + 0,41^2 + 0,5^2 + 0,33^2 + (-0,28)^2 + 0,26^2 +} \\
 &\quad \sqrt{0,38^2 + 0,5^2 + 0,29^2} \\
 &= 1,45489
 \end{aligned}$$

**Tabel 4. 20 Hasil perhitungan  $|V_{ij}|$**

Vij	$ V_{ij} $
Z1	1.45489
Z2	1.35996
Z3	1.4893

- Menghitung nilai  $\beta$  dengan persamaan 2.7  
 $\beta = 0,7(1)^{1/15} = 0,75319$
- Update bobot  $V_{ij}$  dengan menggunakan bobot  $V_{ij}$  lama menggunakan persamaan 2.9 serta menentukan bias dengan range  $[-\beta : \beta]$  yang ditunjukkan pada tabel 4.21

$$V_{11} = \frac{V_{11}(lama)\beta}{||V_{i1}||} = \frac{(-0,45)*0,75319}{1,45489} = -0,23296$$

**Tabel 4. 21 Bobot baru  $V_{ij}$**

Vij	Z1	Z2	Z3
bias	0.45	-0.3	0.5
X1	-0.23296	0.148301	0.111262
X2	0.248495	-0.29111	-0.13149
X3	-0.11907	0.08239	0.166893

X4	0.196725	0.247169	0.262984
X5	0.129424	-0.21971	0.177008
X6	-0.21743	0.153794	0.187123
X7	0.160486	-0.12633	-0.19724
X8	0.212256	-0.12084	0.242754
X9	0.258849	0.131823	-0.22758
X10	0.17084	0.230691	0.171951
X11	0.14496	-0.20323	0.21241
X12	0.134601	0.252661	0.171951
X13	0.196725	-0.13182	0.247811
X14	0.258849	-0.20323	-0.20735
X15	0.150132	0.236183	0.126434

2. Melakukan perhitungan *feed forward*

- Menghitung neuron yang masuk ke *hidden layer* dari *input layer* yang ditunjukkan pada tabel 4.22

$$Z_{netj_{11}} = 0,45 + (0,36*(-0,23296)) + (0,58*0,2485) + (0,5*(0,11907) + (0,9*0,19673) + (0,1*0,12942) + (0,1*(-0,21743)) + (0,63*0,16049) + (0,1*0,21226) + (0,1*0,25885) + (0,1*0,17084) + (0,1*(-0,14496)) + (0,1*0,1346) + (0,1*0,19673) + (0,1*0,25885) + (0,1*0,15023) = 0,84$$

**Tabel 4. 22 Hasil perhitungan Z\_netj Data 1 Iterasi 1**

Z_net1	Z_net2	Z_net3
0.84	-0.21	0.73

- Menghitung neuron yang keluar dari *hidden layer* menuju *output layer* yang ditunjukkan pada tabel 4.23

$$Z_1 = \frac{1}{1+e^{-(0,84)}} = 0,70$$

**Tabel 4. 23 Hasil perhitungan Zi Data 1 Iterasi 1**

Z1	Z2	Z3
0.70	0.45	0.67

- Menghitung neuron yang masuk ke *output layer* dari *hidden layer* yang ditunjukkan pada tabel 4.24

$$Y_{netk_{11}} = 0,5 + (0,36*0,70)+((-0,27)*0,45)+(0,64*0,67) = 0,9003$$

**Tabel 4. 24 Hasil perhitungan  $Y_{netk}$  Data 1 Iterasi 1**

$Y_{netk}$
0.9003

- Menghitung neuron yang masuk ke *output layer* dari *hidden layer* yang ditunjukkan pada tabel 4.25

$$Y_k = \frac{1}{1+e^{-(0,9003)}} = 0,711$$

**Tabel 4. 25 Hasil perhitungan  $Y_k$  Data 1 Iterasi 1**

$Y_k$
0.711

3. Melakukan perhitungan *backpropagation*

- Menghitung nilai *error* antara *hidden layer* dan *output layer* yang ditunjukkan pada tabel 4.26

$$\begin{aligned} \delta_{k1} &= (0,1 - 0,711) * (1 - 0,711) * 0,711 \\ &= -0,126 \end{aligned}$$

**Tabel 4. 26 Hasil perhitungan  $\delta_k$  Data 1 Iterasi 1**

$\delta_k$
-0.126

- Menghitung perubahan bobot antara *hidden layer* dan *output layer* yang ditunjukkan pada tabel 4.27

$$\begin{aligned} \Delta W_{jk} &= \alpha \delta_k Z_i \\ &= 0,2 * (-0,126) * (0,70) = -0,018 \end{aligned}$$

$$\begin{aligned} \Delta W_{bias} &= \alpha \delta_k \\ &= 0,2 * (-0,126) = -0,025 \end{aligned}$$

**Tabel 4. 27 Hasil perhitungan  $\Delta W_{jk}$  Data 1 Iterasi 1**

$\Delta W_{jk}$	$\delta_1$
Z1	-0.018
Z2	-0.011
Z3	-0.017
bias	-0.025

- Menghitung nilai *error* antara *hidden layer* dan *input layer* yang ditunjukkan pada tabel 4.28

$$\delta_{netj} = (-0,126) * 0,36 = -0,045$$

**Tabel 4. 28 Hasil perhitungan  $\delta_{netj}$  Data 1 Iterasi 1**

$\delta_{net1}$	$\delta_{net2}$	$\delta_{net3}$
-0,045	0.034	-0.050

$$\begin{aligned} \delta_j &= (-0,045) * (1 - 0,70) * 0,70 \\ &= -0,010 \end{aligned}$$

**Tabel 4. 29 Hasil perhitungan  $\delta_j$  Data 1 Iterasi 1**

$\delta_1$	$\delta_2$	$\delta_3$
-0.010	0.008	-0.011

- Menghitung perubahan bobot antara *hidden layer* dan *input layer* yang ditunjukkan pada tabel 4.30

$$\begin{aligned} \Delta V_{ij} &= \alpha \delta_j X_{11} \\ &= 0,2 * (-0,010) * (0,36) = -0,0007 \end{aligned}$$

$$\begin{aligned} \Delta V_{bias} &= \alpha \delta_j \\ &= 0,2 * (-0,010) = -0,0019 \end{aligned}$$

**Tabel 4. 30 Hasil perhitungan  $\Delta V_{ij}$  Data 1 Iterasi 1**

$\Delta V_{ij}$	$\delta_1$	$\delta_2$	$\delta_3$
1	-0.0007	0.0006	-0.0008
2	-0.0011	0.0010	-0.0013
3	-0.0010	0.0008	-0.0011
4	-0.0017	0.0015	-0.0020
5	-0.0002	0.0002	-0.0002
6	-0.0002	0.0002	-0.0002
7	-0.0012	0.0011	-0.0014
8	-0.0002	0.0002	-0.0002
9	-0.0002	0.0002	-0.0002
10	-0.0002	0.0002	-0.0002
11	-0.0002	0.0002	-0.0002
12	-0.0002	0.0002	-0.0002
13	-0.0002	0.0002	-0.0002
14	-0.0002	0.0002	-0.0002
15	-0.0002	0.0002	-0.0002
bias	-0.0019	0.0017	-0.0022

#### 4. Tahap perubahan bobot

- Menghitung bobot baru antara *hidden layer* dan *output layer* yang ditunjukkan pada tabel 4.31

$$\begin{aligned}
 W_{jk}(\text{baru}) &= W_{jk}(\text{lama}) + \Delta W_{jk} \\
 &= 0,36 + (-0,018) \\
 &= 0,342
 \end{aligned}$$

**Tabel 4. 31 Hasil perhitungan  $W_{jk}(\text{baru})$  Data 1 Iterasi 1**

Wjk	Y1
Z1	0.342
Z2	-0.281
Z3	0.383
bias	0.475

- Menghitung bobot baru antara *hidden* layer dan *input* layer yang ditunjukkan pada tabel 4.32

$$\begin{aligned}
 V_{ij}(\text{baru}) &= V_{ij}(\text{lama}) + \Delta V_{ij} \\
 &= -0,233 + (-0,0007) \\
 &= -0,234
 \end{aligned}$$

**Tabel 4. 32 Hasil perhitungan  $V_{ij}(\text{baru})$  Data 1 Iterasi 1**

Vij	Z1	Z2	Z3
X1	-0.234	0.150	0.110
X2	0.247	-0.276	-0.133
X3	-0.120	0.084	0.166
X4	0.195	0.251	0.261
X5	0.129	-0.221	0.177
X6	-0.218	0.155	0.187
X7	0.159	-0.126	-0.199
X8	0.212	-0.122	0.243
X9	0.259	0.133	-0.228
X10	0.171	0.233	0.172
X11	-0.145	-0.205	-0.213
X12	0.134	0.255	0.172
X13	0.197	-0.133	0.248
X14	0.259	-0.205	-0.208
X15	0.150	0.238	0.126
bias	0.448	-0.298	0.498

## Data 2

### 1. Melakukan perhitungan *feed forward*

- Menghitung neuron yang masuk ke *hidden* layer dari *input* layer yang ditunjukkan pada tabel 4.33

$$\begin{aligned}
Z_{\text{netj}_{11}} &= 0,448 + (0,1*(-0,234)) + (0,42*0,247) + (0,5*(-0,120) + \\
&\quad (0,9*0,195) + (0,9*0,129) + (0,1*(-0,218)) + (0,63*0,159) + \\
&\quad (0,63*0,212) + (0,1*0,259) + (0,1*0,171) + (0,1*(-0,145)) + \\
&\quad (0,1*0,134) + (0,1*0,197) + (0,1*0,259) + (0,1*0,150) \\
&= 1,08
\end{aligned}$$

**Tabel 4. 33 Hasil perhitungan Z\_netj Data 2 Iterasi 1**

Z_net1	Z_net2	Z_net3
1.08	-0.44	0.98

- Menghitung neuron yang keluar dari *hidden layer* menuju *output layer* yang ditunjukkan pada tabel 4.34

$$Z_1 = \frac{1}{1+e^{-(1,08)}} = 0,75$$

**Tabel 4. 34 Hasil perhitungan Zi Data 2 Iterasi 1**

Z1	Z2	Z3
0.75	0.39	0.73

- Menghitung neuron yang masuk ke *output layer* dari *hidden layer* yang ditunjukkan pada tabel 4.35

$$\begin{aligned}
Y_{\text{netk}_{11}} &= 0,475 + (0,342*0,75)+((-0,281)*0,39)+(0,383*0,73) \\
&= 0,89894
\end{aligned}$$

**Tabel 4. 35 Hasil perhitungan Y\_netk Data 2 Iterasi 1**

Y_netk
0.89894

- Menghitung neuron yang masuk ke *output layer* dari *hidden layer* yang ditunjukkan pada tabel 4.36

$$Y_k = \frac{1}{1+e^{-(0,89894)}} = 0,711$$

**Tabel 4. 36 Hasil perhitungan Yk Data 2 Iterasi 1**

Yk
0.711

### 3. Melakukan perhitungan *backpropagation*

- Menghitung nilai *error* antara *hidden layer* dan *output layer* yang ditunjukkan pada tabel 4.37

$$\begin{aligned}
\delta k_1 &= (0,1 - 0,711) * (1 - 0,711) * 0,711 \\
&= -0,126
\end{aligned}$$



**Tabel 4. 37 Hasil perhitungan  $\delta_k$  Data 2 Iterasi 1**

$\delta_k$
-0.126

- Menghitung perubahan bobot antara *hidden* layer dan *output* layer yang ditunjukkan pada tabel 4.38

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

$$= 0,2 * (-0,126) * (0,75) = -0,019$$

$$\Delta W_{bias} = \alpha \delta_k$$

$$= 0,2 * (-0,126) = -0,025$$

**Tabel 4. 38 Hasil perhitungan  $\Delta W_{jk}$  Data 2 Iterasi 1**

$\Delta W_{jk}$	$\delta_1$
Z1	-0.019
Z2	-0.010
Z3	-0.018
Bias	-0.025

- Menghitung nilai *error* antara *hidden* layer dan *input* layer yang ditunjukkan pada tabel 4.39

$$\delta_{\_netj} = (-0,126) * 0,342$$

$$= -0,043$$

**Tabel 4. 39 Hasil perhitungan  $\delta_{\_netj}$  Data 2 Iterasi 1**

$\delta_{\_net1}$	$\delta_{\_net2}$	$\delta_{\_net3}$
-0.043	0.035	-0.048

$$\delta_j = (-0,043) * (1 - 0,75) * 0,75$$

$$= -0,010$$

**Tabel 4. 40 Hasil perhitungan  $\delta_j$  Data 2 Iterasi 1**

$\delta_1$	$\delta_2$	$\delta_3$
-0.008	0.008	-0.010

- Menghitung perubahan bobot antara *hidden* layer dan *input* layer yang ditunjukkan pada tabel 4.41

$$\Delta V_{ij} = \alpha \delta_j X_{i1}$$

$$= 0,2 * (-0,008) * (0,1) = -0,0002$$

$$\Delta V_{bias} = \alpha \delta_j$$

$$= 0,2 * (-0,010) = -0,0016$$

**Tabel 4. 41 Hasil perhitungan  $\Delta V_{ij}$  Data 2 Iterasi 1**

$\Delta V_{ij}$	$\delta_1$	$\delta_2$	$\delta_3$
1	-0.0002	0.0002	-0.0002
2	-0.0007	0.0007	-0.0008
3	-0.0008	0.0008	-0.0010
4	-0.0015	0.0015	-0.0017
5	-0.0015	0.0015	-0.0017
6	-0.0002	0.0002	-0.0002
7	-0.0010	0.0011	-0.0012
8	-0.0010	0.0011	-0.0012
9	-0.0002	0.0002	-0.0002
10	-0.0002	0.0002	-0.0002
11	-0.0002	0.0002	-0.0002
12	-0.0002	0.0002	-0.0002
13	-0.0002	0.0002	-0.0002
14	-0.0002	0.0002	-0.0002
15	-0.0002	0.0002	-0.0002
bias	-0.0016	0.0017	-0.0019

#### 4. Tahap perubahan bobot

- Menghitung bobot baru antara *hidden* layer dan *output* layer yang ditunjukkan pada tabel 4.42

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

$$= 0,342 + (-0,019)$$

$$= 0,323$$

**Tabel 4. 42 Hasil perhitungan  $W_{jk}(\text{baru})$  Data 2 Iterasi 1**

$W_{jk}$	$Y_1$
Z1	0.342
Z2	-0.291
Z3	0.365
Bias	0.450

- Menghitung bobot baru antara *hidden* layer dan *input* layer yang ditunjukkan pada tabel 4.43

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

$$= -0,234 + (-0,0002) = -0,234$$

**Tabel 4. 43 Hasil perhitungan Vij(baru) Data 2 Iterasi 1**

Vij	Z1	Z2	Z3
X1	-0.234	0.150	0.110
X2	0.247	-0.275	-0.134
X3	-0.121	0.085	0.165
X4	0.195	0.252	0.259
X5	0.128	-0.220	0.175
X6	-0.218	0.155	0.187
X7	0.158	-0.125	-0.200
X8	0.211	-0.121	0.241
X9	0.258	0.133	-0.228
X10	0.170	0.233	0.172
X11	-0.145	-0.205	-0.213
X12	0.134	0.255	0.172
X13	0.196	-0.133	0.247
X14	0.258	-0.205	-0.208
X15	0.150	0.238	0.126
bias	0.446	-0.297	0.496

Perhitungan dilanjutkan sampai data ke 8 dengan cara sama seperti data sebelumnya. Setelah menghitung semua data training, mencari nilai error dari masing-masing data sebagai berikut:

$$Error_1 = 0,1 - 0,711 = 0,189$$

**Tabel 4. 44 Perhitungan nilai error**

Error	
Data	1
1	-0.611
2	-0.611
3	-0.592
4	-0.579
5	0.222
6	0.220
7	0.215
8	0.211

Perhitungan MSE sebagai berikut:

$$\begin{aligned}
 MSE &= \frac{1}{8} * (-0,611)^2 + (-0,611)^2 + (-0,592)^2 + (-0,579)^2 + (0,222)^2 \\
 &\quad + (0,220)^2 + (0,215)^2 + (0,211)^2 \\
 &= 0,20243
 \end{aligned}$$

Nilai 0,20802 > 0,01, maka dilakukan perhitungan untuk iterasi ke-2 dengan tahap seperti iterasi 1 hingga nilai MSE < 0,01 atau maksimum iterasi telah terpenuhi.

### Perhitungan Data Uji

Setelah pelatihan dilakukan terhadap seluruh data training, maka selanjutnya melakukan perhitungan untuk data uji pada Tabel 4.45.

**Tabel 4. 45 Data uji**

Data	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
1	1	4	2	5	3	1	5	1	1	1	1	1	1	1	1
2	2	4	2	5	3	1	3	1	1	1	1	1	1	1	1
3	3	5	2	5	3	3	3	6	5	3	3	5	5	3	2
4	4	5	1	5	3	3	5	6	5	3	3	5	5	3	2

Normalisasi Data:

$$X1 = \frac{1 - 1}{4 - 1} * (0,9 - 0,1) + 0,1 = 0,1$$

Normalisasi dilakukan terhadap seluruh data latih pada Tabel 4.45, sehingga didapatkan hasil normalisasi data pada Tabel 4.46 berikut.

**Tabel 4. 46 Data uji yang telah dinormalisasi**

Data	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	Y
1	0.1	0.58	0.5	0.9	0.9	0.1	0.63	0.36	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2	0.36	0.58	0.5	0.9	0.9	0.1	0.36	0.36	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
3	0.63	0.74	0.5	0.9	0.9	0.5	0.36	0.1	0.9	0.9	0.5	0.5	0.9	0.9	0.5	0.9
4	0.9	0.74	0.1	0.9	0.9	0.5	0.63	0.36	0.9	0.9	0.5	0.5	0.9	0.9	0.5	0.9

#### 1. Melakukan perhitungan *feed forward*

- Menghitung neuron yang masuk ke *hidden layer* dari *input layer* yang ditunjukkan pada tabel 4.47

$$\begin{aligned} Z_{\text{net}j_{11}} &= 0,440 + (0,1 * (-0,233)) + (0,58 * 0,246) + (0,5 * (-0,125) + \\ &\quad (0,9 * 0,189) + (0,9 * 0,123) + (0,1 * (-0,217)) + (0,63 * 0,157) + \\ &\quad (0,36 * 0,210) + (0,1 * 0,260) + (0,1 * 0,172) + (0,1 * (-0,144)) + \\ &\quad (0,1 * 0,135) + (0,1 * 0,198) + (0,1 * 0,2560) + (0,1 * 0,150) \\ &= 1,04 \end{aligned}$$

**Tabel 4. 47 Hasil perhitungan Z-netj Data Uji**

Z_net1	Z_net2	Z_net3
1.04	-0.44	0.88

- Menghitung neuron yang keluar dari *hidden layer* menuju *output layer* yang ditunjukkan pada tabel 4.48

$$Z_1 = \frac{1}{1+e^{-(1,04)}} = 0,74$$

**Tabel 4. 48 Hasil perhitungan Zi Data Uji**

Z1	Z2	Z3
0.74	0.39	0.71

- Menghitung neuron yang masuk ke *output layer* dari *hidden layer* yang ditunjukkan pada tabel 4.49

$$Y_{\text{netk}_{11}} = 0,381 + (0,74*0,289)+(0,39*(-0,318))+(0,71*0,318) \\ = 0,6939$$

**Tabel 4. 49 Hasil perhitungan Y\_netk Data Uji**

Y_netk
0.6939

- Menghitung neuron yang masuk ke *output layer* dari *hidden layer* yang ditunjukkan pada tabel 4.50

$$Y_1 = \frac{1}{1+e^{-(0,57168)}} = 0,667$$

**Tabel 4. 50 Hasil perhitungan Yk Data Uji**

Yk
0.667

2. Melakukan denormalisasi terhadap Yk yang telah didapat

$$\text{Denormalisasi} = \frac{(0,667-0,1)*(2-1)}{0,8} = 1,7085$$

**Tabel 4. 51 Hasil perhitungan denormalisasi**

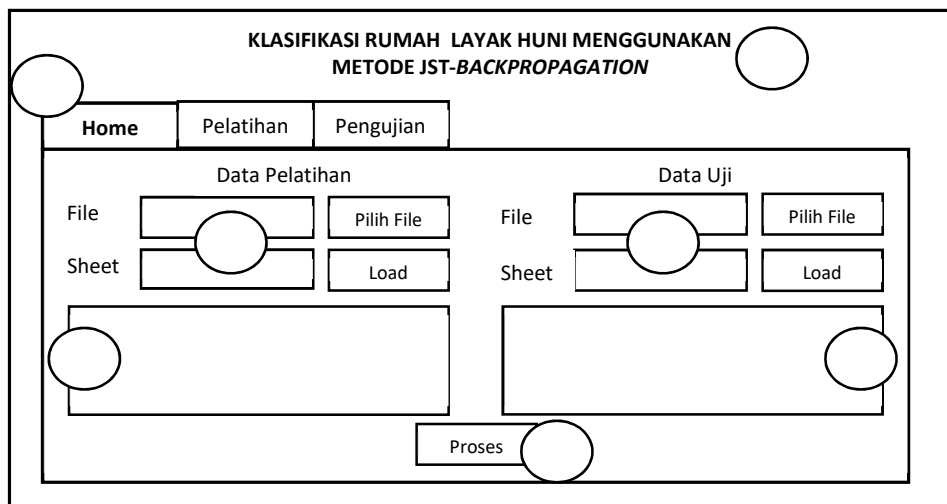
No	Denorm	Target
1	2	1
2	2	1
3	2	2
4	2	2

Perhitungan akurasi:

$$\text{akurasi} = \frac{2}{4} * 100 = 50\%$$

## 4.5 Perancangan Antarmuka

Perancangan antarmuka terdiri dari perancangan tampilan beserta menu yang disediakan sistem. Perancangan tampilan dibagi menjadi 7 bagian yaitu header judul, tab menu, area input file data pelatihan, area input data uji, area tampilan data latih, area tampilan data uji, dan tombol proses. Perancangan antarmuka sistem digambarkan pada Gambar 4.9.



**Gambar 4. 9 Perancangan Antarmuka Sistem**

Tampilan antarmuka sistem pada Gambar 4.9 terdiri dari sebagai berikut:

1. Judul aplikasi
2. Menu yang disediakan oleh aplikasi
3. Input file dan sheet data pelatihan
4. Input file dan sheet data uji
5. Tampilan data pelatihan
6. Tampilan data uji
7. Tombol proses data yang telah dimasukkan

## 4.6 Perancangan Pengujian

Tahap ini melakukan pengujian untuk menghasilkan sistem yang sesuai dengan yang diharapkan. Pengujian dilakukan untuk mengetahui dan menganalisa hasil metode backpropagation dengan inisialisasi bobot *nguyen-widrow*.

Skenario uji diperlukan sebelum melakukan pengujian yang bertujuan agar dapat melakukan pengujian dengan tepat. Berikut skenario pengujian pada klasifikasi rumah layak huni:

1. Jumlah neuron *input layer*

Pengujian jumlah neuron *input* layer dilakukan untuk mengetahui jumlah neuron *input* layer yang tepat untuk mendapatkan hasil klasifikasi yang lebih baik. Pengujian dilakukan dengan mengubah nilai neuron *input* layer. Pengujian ini menggunakan nilai *input* layer sebanyak 15 dan iterasi maksimal sebanyak 100 kali.

## 2. Jumlah neuron hidden layer

Pengujian jumlah neuron *hidden* layer dilakukan untuk mengetahui jumlah neuron *hidden* layer yang tepat untuk mendapatkan hasil klasifikasi yang lebih baik. Pengujian dilakukan dengan mengubah nilai neuron *hidden* layer. Pengujian ini menggunakan nilai *hidden* layer sebanyak 3 dan iterasi maksimal sebanyak 100 kali.

## 3. Pengujian perbandingan jumlah data latih dan uji

Setelah didapatkan arsitektur jaringan syaraf tiruan yang optimal, pola terbaik akan digunakan untuk mengetahui perbandingan jumlah data latih dan data uji. Pengujian dilakukan dengan membandingkan data latih dan data uji dengan 4 jenis perbandingan yaitu:

- 85% data latih : 15% data uji
- 70% data latih : 30% data uji
- 60% data latih : 40% data uji
- 50% data latih : 50% data uji