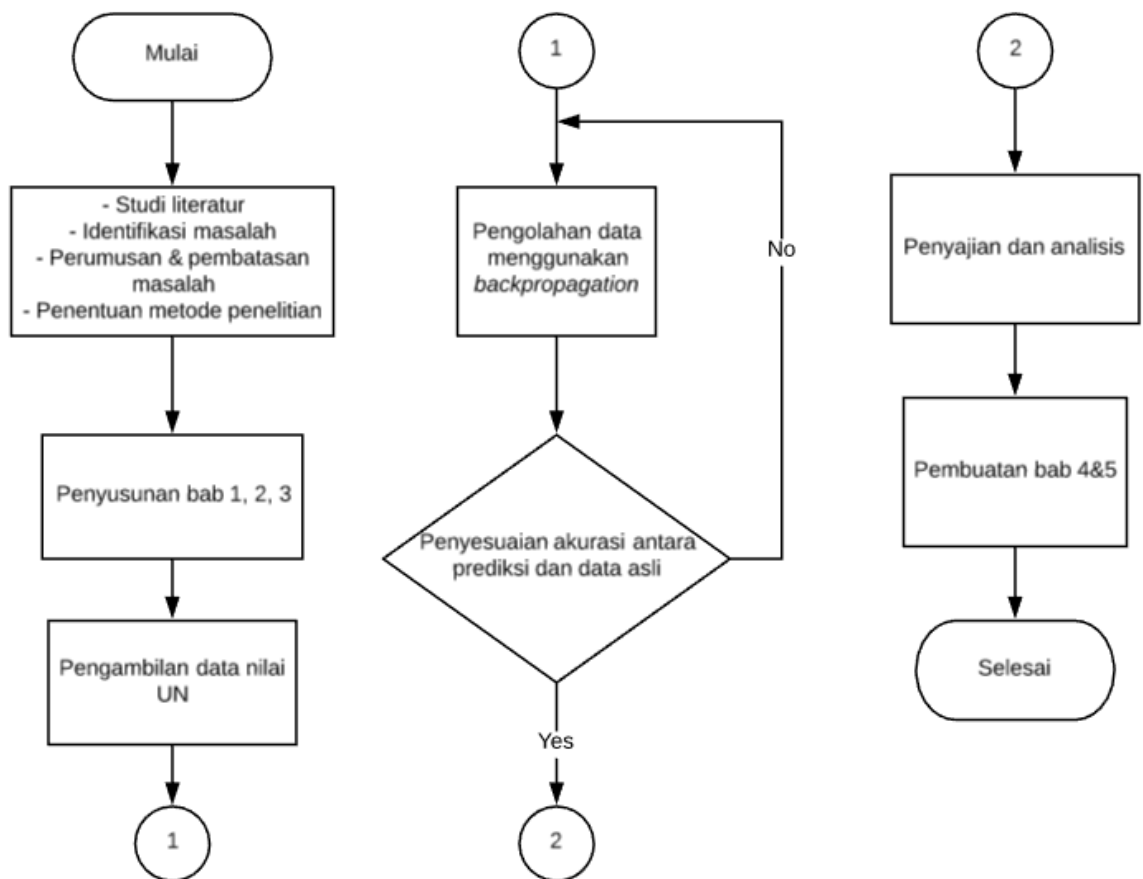


BAB III METODE PENELITIAN

3.1 Prosedur Penelitian

Prosedur yang digunakan pada penelitian ini secara garis besar dapat dilihat pada *flowchart* gambar 3.1. Prosedur penelitian berawal dari studi literatur hingga penyusunan laporan. Penelitian ini menggunakan metode kuantitatif.



Gambar 3.1 Flowchart Prosedur Penelitian

Penelitian ini dimulai dengan tahapan melakukan studi literatur diantaranya membaca dan mempelajari berbagai jurnal dan buku yang berkaitan dengan algoritma *backpropagation* dan jaringan saraf tiruan. Setelah melakukan studi literatur, ditentukan identifikasi masalah, perumusan dan pembatasan masalah dan juga menentukan metode penelitian yang digunakan.

Tahap selanjutnya yaitu mengambil data nilai UN SMK 2 Cimahi selama 6 tahun. Data-data tersebut kemudian diolah menggunakan algoritma *backpropagation* menggunakan *software* Matlab. Data yang telah diprediksi kemudian disesuaikan dengan data asli. Setelah data prediksi diperoleh maka bisa dilakukan analisis dan memerhatikan akurasi.

Penyajian dan analisis hasil diperoleh dari prediksi UN siswa dalam bentuk nilai. Analisis tersebut disajikan dalam bentuk angka, tabel, gambar dan deskripsi. Rekomendasi dibuat untuk memberikan rekomendasi mengenai hal-hal apa saja yang sebaiknya dilakukan terhadap penelitian ini dan penelitian selanjutnya agar hasilnya dapat jauh lebih baik. Kesimpulan dibuat untuk mengungkapkan hasil penelitian ini secara ringkas. Jika semua prosedur penelitian selesai, langkah terakhir yaitu penyusunan laporan sebagai bukti pertanggungjawaban penelitian skripsi.

3.2 Objek Penelitian

Objek penelitian ini yaitu data nilai UN yang diperoleh dari SMKN 2 Cimahi yang berlokasi di Jalan Kamarung No 69, Citeureup, Kecamatan Cimahi Utara, Kota Cimahi 40512. Data yang digunakan merupakan data kelas 12 jurusan mekatronika.

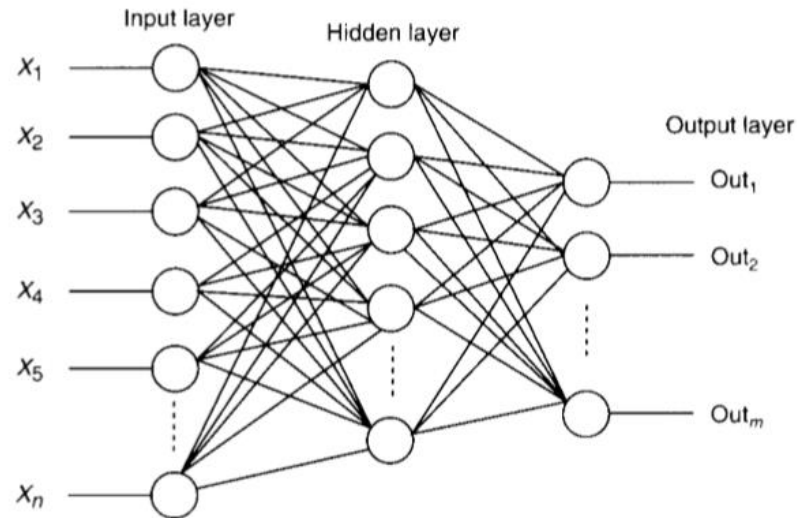
3.3 Pengumpulan Data Nilai Ujian Nasional

Ujian Nasional sangat penting karena Ujian Nasional merupakan salah satu syarat kelulusan. Untuk melakukan prediksi nilai maka yang dibutuhkan adalah data relevan. Penulis menggunakan data nilai selama 6 tahun, dimulai dari tahun 2011 hingga tahun 2016. Data tersebut diolah secara bertahap. Jaringan saraf tiruan telah melakukan prediksi yang digunakan di masa depan. Untuk mendapatkan prediksi pertama maka data harus dilatih terlebih dahulu.

3.4 Tahapan *Backpropagation*

Algoritma *backpropagation* adalah salah satu algoritma jaringan saraf tiruan yang paling banyak digunakan, karena memiliki kemampuan pemasangan non-linier yang sangat tinggi, dan dapat digunakan di banyak bidang. *Backpropagation*

adalah algoritma pembelajaran terawasi dan biasanya digunakan dengan beberapa lapisan untuk mengubah bobot yang terhubung ke jaringan yang ada di dalam lapisan seperti ditunjukkan pada gambar 3.2.



Sumber: (Z. Zacharis, 2016)

Gambar 3.2 Struktur Jaringan *Backpropagation*

Dengan melihat secara mendalam melalui literatur dan penelitian para ahli, sejumlah faktor memiliki efek pada hasil penelitian. Faktor ini dipelajari dan disinkronkan ke dalam angka yang sesuai untuk pengkodean komputer dalam lingkungan jaringan saraf tiruan. Faktor-faktor ini diklasifikasikan sebagai variabel *input* dan variabel *output*. Adapun proses algoritma *backpropagation* yang akan dilakukan dalam proses pelatihan algoritma *backpropagation* adalah menentukan parameter-parameter jaringan *backpropagation* yaitu menentukan *input*, menentukan *output*, menentukan bobot dan menentukan bias. Variabel *input* yang diidentifikasi adalah nilai Ujian Nasional yang berisikan mata pelajaran Bahasa Indonesia, Bahasa Inggris, Matematika dan Kompetensi. Variabel *output* mewakili hasil prediksi dari *output* nilai Ujian Nasional.

Algoritma *backpropagation* untuk melakukan *training* terhadap suatu jaringan terdiri dari tiga tahap, yaitu *feedforward* dari pola *input training*, *backpropagation* dari *error* yang terkait, dan penyesuaian bobot.

a. Tahap 1: *Feedward Propagation*

Pada tahap ini pola *input* dihitung maju dari lapisan *input* ke lapisan *output* menggunakan fungsi aktivasi yang telah ditentukan untuk mendapatkan nilai dari *output* jaringan.

b. Tahap 2: *Backpropagation*

Pada tahap ini, kesalahan perhitungan yaitu perbedaan antara nilai *output* jaringan dengan nilai target yang diinginkan. Nilai tersebut disinkronkan dari garis yang terkait dengan unit-unit di lapisan *output*.

c. Tahap 3: Penyesuaian Bobot

Setelah mendapatkan semua nilai faktor, semua bobot dalam jaringan dimodifikasi berdasarkan nilai faktor. Fase ketiga diulang terus menerus sebagai pola data pelatihan dan diulangi hingga mencapai kondisi berhenti. Kondisi berhenti yang biasa digunakan meliputi jumlah iterasi dan toleransi kesalahan. Variasi model struktur JST diimplementasikan dengan Matlab *Neural Network Tools Box* versi 8.6. Setelah model JST dibangun dan digunakan untuk memprediksi nilai UN, output dibandingkan dengan nilai riil yang diukur oleh *Root Mean-Square Error*, *Mean Absolute Error* dan *Mean Absolute Percentage Error* untuk setiap set data. Langkah terakhir adalah membentuk beberapa alternatif portofolio menggunakan kombinasi variasi dari hasil JST.

3.5 Pelatihan Standar *Backpropagation*

Berikut langkah-langkah pelatihan dengan *backpropagation*:

Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil. *Set learning rate* α ($0 < \alpha \leq 1$)

Langkah 1 : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2-8

Langkah 2 : Untuk setiap pasang data latih, lakukan langkah 3-8

Tahap I : *Feedward Propagation*

Langkah 3 : Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya

Langkah 4 : Hitung semua keluaran di unit tersembunyi z_j ($j = 1, 1, \dots, p$)

$$z_{net_j} = v_{j0} + \sum_{i=1}^p x_i v_{ji}$$

Mengaplikasikan fungsi aktivasi untuk menghitung sinyal *output*

$$z_j = f(z_{in_j})$$

Dan mengirim sinyal ke semua unit di lapisan di atasnya (*output unit*)

Langkah 5 : Hitung semua keluaran jaringan di unit $y_k (k = 1, 2, \dots, m)$

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j W_{jk}$$

dan mengaplikasikan fungsi aktivasinya untuk menghitung sinyal *output*

$$y_k = f(y_{in_k})$$

Tahap II : Backpropagation

Langkah 6 : Hitung faktor δ unit keluaran berdasarkan *error* di setiap unit keluaran $y_k (k = 1, 2, \dots, m)$

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k)$$

δ_k merupakan unit *error* yang akan dipakai dalam perubahan bobot lapisan di bawahnya (langkah 7)

Menghitung koreksi bobotnya (digunakan untuk memperbaharui W_{jk})

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

Menghitung koreksi bias (digunakan untuk memperbaharui W_{0k})

$$\Delta w_{ok} = \alpha \delta_k$$

Langkah 7 : Hitung faktor unit tersembunyi berdasarkan error di setiap unit tersembunyi $z_j (j = 1, 2, \dots, p)$

Faktor δ unit tersembunyi :

$$\delta_j = \delta_{net_j} f'(z_{net_j}) = \delta_{net_j} z_j (1 - z_j)$$

Hitung suku perubahan bobot v_{ji} (yang akan dipakai nanti untuk merubah bobot v_{ji})

$$\Delta v_{ji} = \alpha \delta_j x_i ; j = 1, 2, \dots, p ; i = 0, 1, 2, \dots, n$$

Tahap III : Penyesuaian Bobot

Langkah 8 : Hitung semua perubahan bobot

Perubahan bobot garis yang menuju ke unit tersembunyi :

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} (k = 1, 2, \dots, m ; j = 0, 1, 2, \dots, p)$$

Perubahan bobot garis yang menuju ke unit tersembunyi :

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} (j = 1, 2, \dots, p ; i = 0, 1, 2, \dots, n)$$

Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, hanya propagasi maju (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan.

Langkah 9: Uji kondisi berhenti (akhir iterasi).

3.6 Optimalitas Arsitektur *Backpropagation*

Masalah utama yang dihadapi dalam *backpropagation* adalah lamanya iterasi yang harus dilakukan. *Backpropagation* tidak dapat memberikan kepastian tentang berapa *epoch* yang harus dilalui untuk mencapai kondisi yang diinginkan. Oleh karena itu orang berusaha meneliti bagaimana parameter-parameter jaringan dibuat sehingga menghasilkan jumlah iterasi yang relatif lebih sedikit.

Tujuan utama penggunaan *backpropagation* adalah mendapatkan keseimbangan antara pengenalan pola pelatihan secara benar dan respon yang baik untuk pola lain yang sejenis (disebut data pengujian). Jaringan dapat dilatih terus menerus hingga semua pola pelatihan dikenali dengan benar. Akan tetapi itu tidak menjamin jaringan akan mampu mengenali pola pengujian dengan tepat. Jadi tidaklah bermanfaat untuk meneruskan iterasi hingga semua kesalahan pada pelatihan = 0.

Umumnya data dibagi menjadi 2 bagian, yaitu pola data yang dipakai sebagai pelatihan dan data yang dipakai untuk pengujian. Perubahan bobot dilakukan berdasarkan pola pelatihan, kesalahan yang terjadi dihitung berdasarkan semua data (pelatihan dan pengujian). Selama kesalahan ini menurun, pelatihan terus dijalankan. Akan tetapi, jika kesalahannya meningkat, pelatihan tidak ada gunanya untuk diteruskan lagi. Jaringan sudah mulai mengambil sifat yang hanya dimiliki secara spesifik oleh data pelatihan (tapi tidak dimiliki oleh data pengujian) dan sudah mulai kehilangan kemampuan melakukan generalisasi.

3.7 *Matrix Laboratory* (Matlab)

Dalam memprediksi nilai Ujian Nasional di SMKN 2 Cimahi diperlukan suatu *software* yang dapat membantu mengimplementasikan deteksi dini menjadi sebuah angka-angka yang selanjutnya dapat diolah menjadi suatu informasi. Dalam penelitian ini, penulis menggunakan *software Matrix Laboratory* (Matlab) untuk mencari penyelesaian masalah pengenalan pola dari suatu data yang telah disiapkan. Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu model yang sangat mudah dan penyelesaiannya diekspresikan dalam notasi matematika.

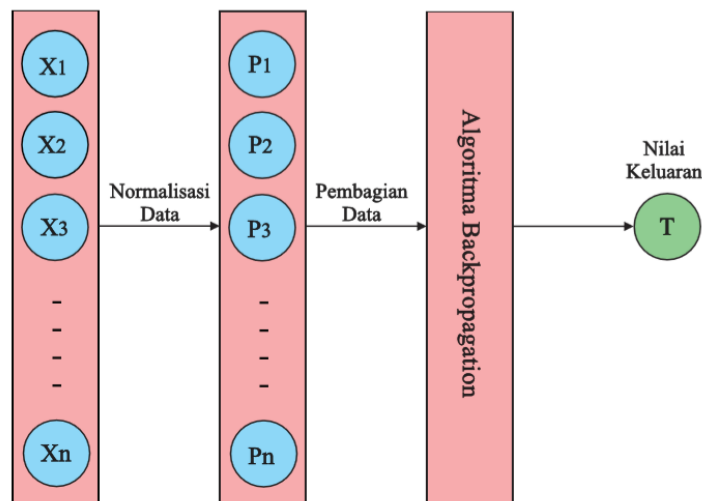
Software Matlab mempunyai beberapa perintah yang dapat digunakan untuk membuat jaringan saraf tiruan *backpropagation*. Perintah-perintah tersebut adalah:

- a. `newff` : digunakan untuk membentuk jaringan saraf dengan *backpropagation*
- b. `train` : digunakan untuk melakukan pelatihan jaringan

- c. *sim* : digunakan untuk melakukan *output* dari jaringan yang sudah dilatih
- d. *minmax* : digunakan untuk menentukan nilai minimum dan maksimum data input
- e. *trainParam* : digunakan untuk menentukan parameter pelatihan jaringan seperti *epoch*, *goal*, *learning rate*, dan *momentum coefisient* (*mc*)
- f. *tansig*, *logsig* : parameter fungsi aktivasi dari suatu layer jaringan. *Tansig* adalah fungsi aktivasi bipolar sigmoid dan *logsig* fungsi aktivasi binary sigmoid.

3.8 Tahapan Sistem Nilai Ujian Nasional

Tahapan umum untuk mengevaluasi nilai Ujian Nasional dapat dilihat pada gambar 3.2 berikut.



Gambar 3.3 Diagram Alur Sistem

Data masukan yang diperlukan yaitu 4 nilai mata pelajaran yaitu Matematika, Bahasa Indonesia, Bahasa Inggris dan Kompetensi. Data masukan kemudian dinormalisasi ke nilai yang terletak antara 0 dan 1. Kemudian dilakukan algoritma *backpropagation* untuk mendapatkan nilai keluaran. Nilai keluaran yang dihasilkan berupa nilai perhitungan Ujian Nasional pada tahun ke-n.

3.9 Representasi Masalah

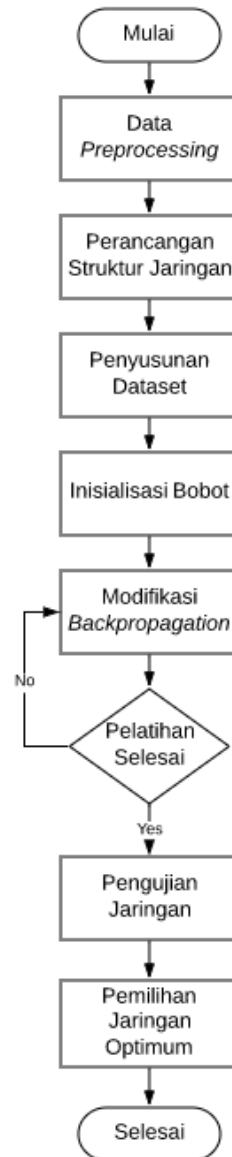
Dalam penelitian mengenai prediksi nilai Ujian Nasional dipengaruhi oleh beberapa parameter. Parameter tersebut adalah 4 mata pelajaran yang terdiri dari Matematika, Bahasa Indonesia, Bahasa Inggris dan Kompetensi.

3.10 Pengembangan Sistem Nilai Ujian Nasional

Dalam pengembangan sistem nilai Ujian Nasional menggunakan algoritma *backpropagation*, ada beberapa tahap yang perlu dilakukan, yaitu sebagai berikut.

- a. *Data preprocessing*
- b. Perancangan struktur jaringan
- c. Penyusunan data set pelatihan dan pengujian
- d. Inisialisasi bobot
- e. Modifikasi algoritma pelatihan *backpropagation*
- f. Pengujian jaringan
- g. Pemilihan jaringan optimum untuk sistem nilai Ujian Nasional

Tahapan-tahapan diatas merupakan suatu proses yang dapat berulang. Proses tersebut dapat dilihat pada *flowchart* gambar 3.3. Sedangkan untuk masing-masing tahapan dijelaskan pada poin 3.10.1 sampai 3.10.7.



Gambar 3.4 *Flowchart* Pengembangan Sistem

3.10.1 Data Preprocessing

Data *preprocessing* dilakukan agar data yang akan digunakan dalam proses evaluasi benar, tidak mengandung kesalahan dan sesuai dengan algoritma yang digunakan. Adapun tahapan yang dilakukan yaitu sebagai berikut.

- a. Data *cleansing* (pembersihan data) yang didalamnya terdapat penanganan data yang hilang dan pendeteksian *outlier*.
- b. Normalisasi data

c. Pembagian data

Penjelasan lebih lanjut mengenai data preprocessing dapat dilihat pada poin 3.10.1.1 sampai 3.10.1.2

3.10.1.1 Data Cleansing

Pada data cleansing terdapat dua tahap yang perlu dilakukan, yaitu:

a. Penanganan Data yang Hilang

Dalam mendeteksi data yang hilang, dapat dilihat pada 4 mata pelajaran yang ada ada lampiran. Bila ada data yang hilang, maka pada data akan terlihat kosong. Apabila terjadi kekosongan pada data tersebut, data yang hilang akan diganti dengan nilai rata-rata dari setiap variabel data.

b. Pendeteksian *Outlier*

Data outlier adalah data yang dianggap memiliki sifat yang berbeda dibandingkan dengan keseluruhan data. Data outlier yang dimaksud adalah data yang kemungkinan salah pada saat proses pemasukan data, misalkan data yang bernilai minus dan data yang jumlahnya tidak normal.

3.10.1.2 Normalisasi Data

Normalisasi dimaksudkan untuk menyeragamkan sifat data dari semua label yang ada. Dikarenakan beberapa label data yang ada memiliki satuan nilai yang berbeda dari label lainnya maka diperlukan keseragaman data agar hasil klasifikasi untuk sistem nilai Ujian Nasional menggunakan algoritma *backpropagation* bisa lebih optimal.

3.10.1.3 Pembagian Data

Langkah terakhir data *preprocessing* yaitu membagi data menjadi dua bagian, yaitu data latih dan data uji. Pelatihan menggunakan data dari tahun 2011-2016.

Dengan data latih, data diuji untuk melihat kemampuan jaringan dalam mengenali pola data yang diberikan. Sedangkan data uji digunakan untuk melihat kemampuan jaringan dalam mengklasifikasikan nilai ujian nasional.

3.10.2 Perancangan Struktur Jaringan

Setelah data *preprocessing*, selanjutnya dilakukan perancangan struktur jaringan. Pada perancangan struktur jaringan terdapat tahap perancangan arsitektur jaringan dan menentukan fungsi aktivasi yang dijabarkan pada poin 3.10.2.1 berikut ini.

3.10.2.1 Menentukan Fungsi Aktivasi

Dalam *backpropagation*, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu: kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut adalah fungsi sigmoid biner yang memiliki range (0,1).

$$f(x) = \frac{1}{1+e^{-x}} \text{ dengan turunan } f'(x) = f(x)(1 - f(x))$$

3.10.3 Menyusun Data Set Pelatihan dan Pengujian

Seperti yang telah dijelaskan sebelumnya, untuk dapat prediksi nilai Ujian Nasional tahun ke-n dipengaruhi oleh data nilai setiap mata pelajaran pada tahun 2010 sampai 2018. Hal ini berpengaruh terhadap set pelatihan yang digunakan. Data set yang digunakan adalah sebagai berikut.

3.10.4 Inisialisasi Bobot

Untuk inisialisasi bobot lapisan tersembunyi, diinisialisasi dengan nilai acak yang cukup kecil biasanya terletak antara 0 dan 1 atau -1. Dalam penelitian ini penulis menggunakan inisialisasi dalam interval 0 dan 1.

Jika a adalah data minimum dan b adalah data maksimum, transformasi linier yang dipakai untuk mentransformasikan data ke interval $[0,1]$ adalah:

$$x' = \frac{0,8(x - a)}{b - a} + 0,1$$

3.10.5 Modifikasi Algoritma Pelatihan *Backpropagation*

Setelah tahapan poin 4.3.1 sampai 4.3.4 dilakukan, tahapan selanjutnya yaitu proses pelatihan untuk mendapatkan bobot dengan kesalahan minimum. Algoritma yang digunakan yaitu *backpropagation*.

3.10.6 Pengujian Jaringan

Untuk mengukur kehandalan jaringan yang telah dilatih maka diperlukan pengujian, pada proses pengujian, arsitektur jaringan sama dengan arsitektur pada saat pelatihan. Bobot yang digunakan pun merupakan bobot hasil pelatihan. Karena pada saat pengujian hanya diperlukan nilai keluaran, maka hanya dilajukan fase propagasi maju.

3.10.7 Pemilihan Jaringan Optimum untuk Klarifikasi

Hasil pelatihan dan pengujian dengan berbagai kasus data inisialisasi yang berbeda-beda, dipilih jaringan dengan nilai MSE data latih terkecil. Perlu diperhatikan juga nilai MSE data latih. Bila nilai MSE data pelatihan rendah namun akurasi yang dihasilkan pada data pengujian rendah, berarti jaringan terlalu banyak belajar (*overfitting*). Jaringan tersebut tidak dipilih sebagai jaringan optimum. Maka jaringan optimum yang dipilih adalah jaringan dengan nilai MSE data pelatihan rendah dan akurasi yang dihasilkan pada data pengujian tinggi.