

### BAB III

#### METODE PENELITIAN

Pada penelitian ini, akan dilakukan beberapa langkah untuk mencapai tujuan dari penelitian tersebut. Penelitian ini akan dimulai dengan pengumpulan data. Adapun data yang digunakan adalah data fluks gelombang radio pada  $\lambda=10,7$  cm (F10.7), data hujan meteor, dan data frekuensi maksimum lapisan E Sporadis (foEs). Data tersebut di atas akan dipisahkan ke dalam dua bagian, yaitu data untuk pelatihan dan data untuk pengujian.

Tahap selanjutnya adalah menentukan desain jaringan. Pada tahap ini akan ditentukan defenisi masalah yaitu penentuan pola masukan dan keluaran untuk keperluan pelatihan dan pengujian pada Jaringan Saraf Tiruan (JST). Tahap ini kemudian diikuti dengan penentuan algoritma pelatihan.

Tahap berikutnya adalah penentuan arsitektur jaringan berdasarkan desain jaringan yang telah didefinisikan. Setelah itu, dilakukan inisialisasi terhadap jaringan yang akan dilatih dan diuji.

Selanjutnya adalah tahap pelatihan terhadap jaringan saraf tiruan menggunakan data yang telah diklasifikasikan dan ditentukan arsitekturnya. Tujuan dari pelatihan ini adalah menentukan nilai SSE (*Sum square error*), nilai epoch, jumlah neuron layer tersembunyi dan bilangan belajar. Metode pelatihan untuk JST yang digunakan dalam penelitian ini adalah *backproagation*. Algoritma pelatihan ini dirancang untuk memperkecil MSE (*mean square error*) antara input sebenarnya dengan output yang diinginkan.

Setelah tahap pelatihan berikutnya adalah tahap pengujian terhadap data pengujian. Tujuan dari tahap pengujian ini adalah untuk mengetahui apakah jaringan saraf tiruan mampu melakukan prediksi dengan baik.

Perangkat lunak yang digunakan dalam perancangan sistem pada penelitian ini adalah dengan menggunakan *MATLAB V7.0.4*, karena memiliki bahasa tingkat tinggi dan dapat digunakan untuk komputasi teknik, penghitungan, visualisasi dan pemrograman. Selain itu juga memiliki *neural network* (NN) *toolbox*, sehingga memudahkan dalam perancangan program JST maupun simulasi dari sistem yang telah dilatih.

### 3.1 Data

Penelitian menggunakan tiga macam data antara lain:

1. Data fluks gelombang matahari pada panjang gelombang  $\lambda=10,7$  cm (F10.7). F10.7 yang diterima di permukaan bumi sangatlah banyak bila dibandingkan dengan data intensitas EUV. Disamping itu korelasi antara aktivitas matahari dengan intensitas sinar ultraviolet sangat tinggi. Lean (1987) memperlihatkan bahwa meskipun variasi luminositas total matahari berorde 0.1 % saja pada daerah cahaya tampak, tetapi variasi spektral pada rentang ultraviolet dapat mencapai orde beberapa ratus persen. Variasi spektral ini berkorelasi tinggi dengan tingkat aktivitas matahari. Pertimbangan lain yang mendukung pemilihan F10.7 sebagai indikator aktivitas matahari adalah Kane (1992) yang menunjukkan bahwa untuk prediksi jangka pendek menggunakan F10.7 masih lebih baik dibandingkan menggunakan bilangan bintik matahari (*sunspot*, *R*).

Data fluks F10.7 diperoleh dari *Solar Geophysical Data* yang diterbitkan secara rutin sebulan sekali dan memuat data matahari dari berbagai stasiun pengamatan di permukaan bumi. Data tersebut dapat juga diperoleh dari internet yang dipublikasikan oleh *Dominion Radio Astrophysical Observatory* di Penticton Kanada. Satuan F10.7 adalah dalam  $10^{-22}$  Watt  $m^{-2}$   $Hz^{-1}$  atau Jansky suatu satuan baku pengukuran radio astronomi.

2. Data hujan meteor (Zenithal Hourly Rate, ZHR). Data ini diperoleh dari IMO (*International Meteor Organization*). Jiyo dan Soureka (1994) telah meneliti gejala ionosfer pada saat terjadinya hujan meteor Eta Aquarid. Meteor yang menembus ionosfer pada kecepatan 10-70 km/ jam menyebabkan terjadinya ionisasi pada ketinggian 80-100 km di atas permukaan bumi. Terjadinya perubahan ionisasi secara tiba-tiba pada lapisan E dapat terjadi saat hujan meteor (Mc Kenley dan Milman, 1991).
3. Data frekuensi maksimum lapisan E Sporadis (foEs). Data ini merupakan frekuensi tertinggi pada lapisan ini. Harga foEs dapat dibaca dari ionogram hasil pengamatan dengan ionosonda variabel tipe IPS42/DBD43. Ionogram yang dihasilkan dibaca dengan peralatan *scaling* yang disebut *Scale-49*. Data dalam penelitian ini data foEs yang digunakan memiliki tingkat kesalahan kurang dari 0,5 MHz. Data ini diperoleh dari Stasiun Pengamatan Dirgantara Tanjungsari Sumedang yang sudah tersedia di Lembaga Penerbangan dan Antariksa Nasional, Bidang Ionosfer dan Telekomunikasi.

Ada dua parameter data fisis yang mempengaruhi munculnya lapisan E sporadis yaitu F10.7 dan Hujan Meteor. Sedangkan untuk parameter keluaran adalah data frekuensi maksimum lapisan E sporadis yang diperoleh dari pengamatan di stasiun Pengamat Dirgantara Tanjungsari (SPDTS). Data pengamatan yang digunakan adalah data harian yang berurutan dengan panjang minimum 7 hari. Dari data harian yang diamati selama 4 tahun didapatkan rangkaian data sebagai berikut.

Tabel 3.1. Tanggal pengamatan data foEs yang dilakukan

Bulan	Pukul 18.00	Pukul 19.00	Pukul 20.00	Pukul 21.00
Agustus 2005	1-7 (7 hari)	2-31(30 hari)	6-30(25 hari)	8-28(21 hari)
April 2006	1-30(30 hari)	1-30(30 hari)	1-30(30 hari)	24-30(7 hari)
Mei 2007	1-30(30 hari)	1-30(30 hari)	1-30(30 hari)	1-30(30 hari)
Januari 2008	1-31(31 hari)	1-28(28 hari)	1-28(28 hari)	5-30(26 hari)

Data pengamatan tahun 2005 dan tahun 2006 dianggap mewakili data pada saat matahari aktif ( $F10.7 > 90$ ). Sedangkan data tahun 2007 dan tahun 2008 mewakili data saat matahari tenang ( $F10.7 < 90$ ). Waktu pengamatan adalah pukul 18.00 hingga pukul 21.00 WIB dengan alasan bahwa pada saat-saat itulah intensitas sinar matahari mencapai minimum sehingga pengaruhnya terhadap ionisasi juga minimum.

Rangkaian data yang digunakan untuk proses pelatihan adalah data foEs hasil pengamatan selama bulan Mei 2007 antara pukul 18.00 WIB sampai pukul 21.00 WIB waktu setempat, dengan alasan bahwa data pada bulan ini adalah data

yang paling lengkap. Data untuk pengujian dikelompokkan menjadi dua. Kelompok pertama adalah data pada bulan Agustus 2005, April 2006, dan Januari 2008 dalam selang waktu yang sama. Kemudian kelompok kedua adalah data pengamatan pada bulan Mei 2007 itu sendiri. Jika data pengamatan pada pukul 18.00 WIB digunakan sebagai data latih, maka data pengamatan pukul 19.00 WIB, pukul 20.00 WIB, dan pukul 21.00 WIB digunakan sebagai data uji. Sebaliknya jika data pengamatan pukul 19.00 WIB digunakan sebagai data latih, maka data pengamatan pukul 18.00 WIB, pukul 20.00 WIB, dan data pukul 21.00 WIB digunakan sebagai data uji. Demikian seterusnya untuk data uji yang lainnya.

### **3.2 Desain Jaringan**

Desain jaringan dilakukan dengan tujuan untuk mengetahui pola masukan dan keluaran. Pola masukan ditentukan berdasarkan parameter berupa data-data yang mempengaruhi munculnya lapisan E sporadis secara fisis. Sedangkan untuk pola keluaran ditentukan berdasarkan data lapisan E sporadis yang diperoleh dari hasil pengamatan. Sebanyak empat belas data yang menjadi data masukannya kemudian data tersebut akan terhubung dengan empat belas neuron pada lapisan input yang kemudian akan terhubung dengan lapisan selanjutnya.

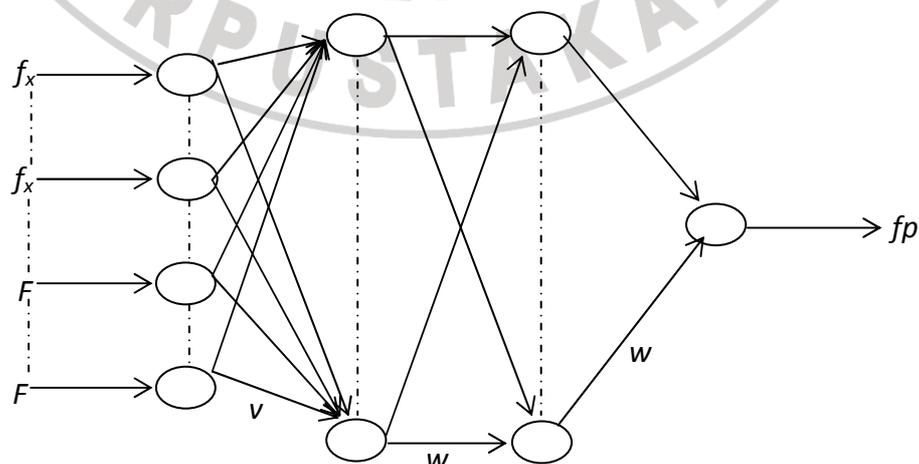
### 3.3 Arsitektur Jaringan

Pada penelitian ini akan dikembangkan tiga arsitektur jaringan yang dibedakan berdasarkan konfigurasi dari jaringan saraf tiruannya. Data masukan akan dinormalkan terlebih dahulu.

Sistem jaringan pertama (A-I) terdiri dari masukan, dua lapisan tersembunyi, dan lapisan keluaran. Jumlah masukan sebanyak empat belas yang terdiri dari tujuh data foEs dan tujuh data F10.7 hasil pengamatan tujuh hari kebelakang. Masing-masing neuron terhubung dengan empat belas neuron pada lapisan tersembunyi pertama dan lapisan tersembunyi pertama terhubung dengan lapisan tersembunyi kedua, sedangkan lapisan tersembunyi kedua terhubung dengan lapisan keluaran.

Kemudian untuk jaringan kedua (A-II) sama dengan jaringan yang pertama hanya dibedakan pada data masukannya saja. Data masukannya adalah data hujan meteor dan data F10.7. Sedangkan untuk jaringan yang ketiga (A-III) yang menjadi data masukan adalah frekuensi maksimum lapisan E sporadis(foEs).

Adapun arsitektur JST dalam penelitian kali ini dapat dilihat pada gambar 3.2 sebagai berikut.



Gambar 3.1. Arsitektur JST

### 3.3.1. Fungsi Aktivasi

Fungsi aktivasi yang digunakan dalam penelitian ini adalah fungsi aktivasi sigmoid biner. Fungsi sigmoid biner memiliki nilai range antara 0 dan 1 sehingga cocok digunakan untuk jaringan saraf yang membutuhkan nilai output antara 0 dan satu. Fungsi sigmoid biner dirumuskan sebagai:

$$O_j = \left( \sum_{i=1}^n W_{ji} X_i \right) \dots \dots \dots (3.1)$$

Dengan  $W_{ji}$  adalah bilangan bobot,  $X_i$  adalah masukan, dan  $F$  adalah fungsi sigmoid berikut:

$$F(x) = \frac{1}{1 + e^{-\alpha x}} \dots \dots \dots (3.2)$$

Salah sifat dari fungsi sigmoid adalah bahwa fungsi turunannya dapat dituliskan sebagai berikut:

$$F'(x) = \sigma F(x)[1 - F(x)] \dots \dots \dots (3.3)$$

Misalkan  $x = X_1 + X_2 + \dots + X_n$  maka dapat ditunjukkan bahwa:

$$\frac{\partial F(x)}{\partial X_1} = \frac{\partial F(x)}{\partial X_2} = \dots = \frac{\partial F(x)}{\partial X_n} = \sigma F(x)[1 - F(x)] \dots \dots \dots (3.4)$$

Selanjutnya misalnya  $x = X_1 Y_1 + X_2 Y_2 + \dots + X_n Y_n$ , maka dapat ditunjukkan pula bahwa:

$$\frac{\partial F(x)}{\partial X_1} = \sigma F(x)[1 - F(x)] Y_1 \dots \dots \dots (3.5)$$

Dan

$$\frac{\partial F(x)}{\partial Y_1} = \sigma F(x)[1 - F(x)] X_1 \dots \dots \dots (3.6)$$

Keluaran  $O_j$  dari persamaan 3.1 dapat menjadi masukan neuron yang lainnya, misalkan neuron tersebut berindeks k, maka dengan rumus 3.1 didapatkan keluaran:

$$O_k = \left( \sum_{i=1}^n W_{ki} O_i \right) \dots \dots \dots (3.7)$$

Dengan m adalah jumlah informasi yang masuk ke dalam neuron k.

Dari sifat-sifat fungsi sigmoid tersebut maka didapatkan turunan fungsi aktivasi F adalah sebagai berikut:

$$O_k = \left( \sum_{i=1}^n W_{ki} O_i \right) = O_k (1 - O_k) O_j \dots \dots \dots (3.8)$$

### 3.3.2. Pelatihan Sistem

Proses pelatihan dalam jaringan saraf tiruan *backpropagation* yang akan diterapkan adalah dengan iterasi memperbaharui harga bilangan bobot W menggunakan teknik *gradient descent dengan momentum* dengan perambatan galat dari neuron keluaran bergerak mundur ke arah neuron antara. Jadi proses pelatihan jaringan neural *backpropagation* adalah mengikutsertakan iterasi sebagai berikut:

$$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji}(t) \dots \dots \dots (3.9)$$

Dengan  $W_{ji}(t+1)$  adalah bilangan bobot yang diperbaharui dengan suku pelatihan:

$$\Delta W_{ji}(t) = -\eta \frac{\partial E}{\partial W_{ji}(t)} = \eta \delta_j O_j \dots \dots \dots (3.10)$$

Dengan E adalah galat pada suatu unit neuron dan  $\eta$  adalah bilangan belajar (*learning rate*) yang didapatkan secara coba-coba dari pengalaman, dan  $\delta_j$  adalah gradien galat. Lippman (1987), Yang dan Bhargava (1993), Li Min Fu (1994),

Rose (1995) menggunakan harga  $\eta$  antara nol dan satu. Namun persamaan gradient descent memberikan syarat bahwa  $\eta$  adalah bilangan kecil yang lebih besar dari nol (Hecht-Nielsen, 1990). Untuk neuron keluaran biasanya digunakan kriteria galat (*error*) yang ditentukan dengan rumus berikut:

$$E = \frac{1}{2}(T_k - O_k)^2 \dots\dots\dots(3.11)$$

$T_k$  adalah harga yang diinginkan dan dalam hal ini berupa data yang didapatkan dari pengamatan.

### 3.3.3. Kriteria Kekonvergenan

Pada tahap pelatihan sistem jaringan saraf tiruan *backpropagation* iterasi diulangi sehingga bobot terakhir sudah tidak ada pembaharuan atau dengan kata lain harga suku pelatihan sama dengan nol. Kondisi seperti ini dikatakan bahwa iterasi sudah mencapai kejenuhan atau tingkat konvergen telah dicapai. Oleh karena harga bilangan belajar  $\eta$  positif dan tidak sama dengan nol, kondisi konvergen terjadi jika gradien galat dari masing-masing neuron berharga nol.

Dengan mengacu kepada kriteria kekonvergenan berdasarkan gradien galat tersebut di atas dan jika dalam satu jaringan neural terdiri dari  $M$  neuron, maka kriteria kekonvergenan dapat dirumuskan sebagai berikut:

$$\frac{1}{M} \sum_{i=1}^M |\delta_i| \leq \beta \dots\dots\dots(3.12)$$

Dengan  $\beta$  adalah bilangan positif yang cukup kecil. Kondisi ini adalah konsepsi rataan gradien galat dari suatu jaringan saraf tiruan sama dengan nol dan dapat diterapkan dalam bentuk program komputer. Jika kondisi tersebut terpenuhi, maka

gradien galat dari masing-masing neuron juga bernilai sangat kecil dan proses iterasi dapat dianggap konvergen.

Laju kekonvergenan iterasi dapat dipercepat dengan menambah suku momentum pada ruas kanan sebagai berikut:

$$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji}(t) + \alpha \Delta W(t-1) \dots \dots \dots (3.13)$$

Dengan  $\alpha$  adalah bilangan momentum. Untuk itu, agar suatu jaringan saraf tiruan yang diuji dapat dibandingkan dengan jaringan saraf tiruan yang lainnya, maka untuk semua jaringan saraf tiruan yang dikembangkan menggunakan harga  $\alpha$  yang sama yaitu 0,8.

#### 3.3.4. Validitas

Dalam proses pelatihan diperlukan data latih yang kontinu sehingga generalisasi bilangan bobot dapat dicapai. Dalam satu rangkaian data latih yang sinambung tujuh data digunakan untuk masukan dan data ke delapan digunakan sebagai data target. Proses ini digeser satu data kedepan untuk proses data selanjutnya sehingga mencakup seluruh data latih. Bilangan bobot hasil pelatihan jaringan saraf tiruan ini kemudian diterapkan untuk data yang lain sebagai data uji. Seperti halnya rangkaian data yang diperlukan pada proses pelatihan jaringan saraf tiruan, maka data uji yang diperlukan adalah delapan data. Tujuh data digunakan sebagai masukan dan data ke delapan digunakan sebagai pembanding. Misalnya harga  $f_o$ Es hasil prediksi adalah  $f_p$  dan data pengamatan adalah  $f_o$ . Defenisi selisih antara  $df = f_o - f_p$ . Dari sejumlah pengujian yang dapat dilakukan akan diperoleh rataan  $df$ . dengan melihat harga-harga dari parameter tersebut

dapat diketahui ketelitian hasil prediksi foEs menggunakan jaringan saraf tiruan yang dikembangkan.

### 3.4 Inisialisasi Jaringan

Inisialisasi jaringan adalah pengenalan gambaran hubungan antar lapisan yang digunakan dalam proses pelatihan jaringan. Setiap unit sel pada satu lapisan dihubungkan secara penuh terhadap sel-sel unit pada lapisan didepannya sehingga akan ditemukan bobot dan bias dari hubungan antara lapisan tersebut. Hubungan antara lapisan yang satu dengan lapisan yang lainnya sudah dijelaskan pada arsitektur jaringan pada sub bab sebelumnya.

Algoritma pelatihan Jaringan saraf tiruan (JST) dengan menggunakan metode *backpropagation* adalah sebagai berikut :

**Langkah 0:** Inisialisasi bobot

**Langkah 1:** Jika kondisi henti tidak terpenuhi, lakukan langkah 2 – 9

**Langkah 2:** Untuk setiap pola input, lakukan step 3 – 8

**Feedforward (perambatan Maju):**

**Pelatihan pada layer 1**

**Langkah 3:** Setiap neuron masukan ( $X_i$   $i= 1...n$ ) menerima sinyal masukan  $x_i$  dan menyebarkannya ke semua neuron pada lapis tersembunyi.

**Langkah 4:** Setiap neuron pada lapis tersembunyi-1 ( $Z_j$ ,  $j=1...p$ ) jumlahkan bobot-bobot sinyal masukan,

$$z_{in_j} = \sum_{i=1}^n x_i v_{ij} \quad (3.1)$$

dan menerapkannya pada fungsi aktivasi untuk menghitung sinyal keluarannya,

$$z_j = f(z\_in_j) \quad (3.2)$$

dan mengirim sinyal ini ke semua neuron pada lapis keluaran,

Setiap neuron pada lapis tersembunyi-2 ( $Z'_j$ ,  $j=1\dots m$ ) jumlahkan bobot-bobot sinyal masukan,

$$z'\_in_j = \sum_{i=1}^n x_i v_{ij} \quad (3.3)$$

dan menerapkannya pada fungsi aktivasi untuk menghitung sinyal keluarannya,

$$z'_j = f(z\_in_j) \quad (3.4)$$

dan mengirim sinyal ini ke semua neuron pada lapis keluaran

**Langkah 5:** Setiap neuron keluaran ( $Y_k$ ,  $k=1\dots o$ ) jumlahkan bobot sinyal yang masuk,

$$y\_in_k = \sum_{j=1}^p z'_j w_{jk} \quad (3.5)$$

dan mengaplikasikan fungsi aktivasinya untuk menghitung sinyal yang akan dikeluarkannya

$$y_k = f(y\_in_k) \quad (3.6)$$

#### ***Backpropagation error***

**Langkah 6:** Setiap neuron keluaran ( $Y_k$ ,  $k=1\dots o$ ) menerima sebuah pola target yang berhubungan dengan pola masukan pelatihan dan menghitung kesalahan informasi

$$\delta_k = (t_k - y_k) f'(y\_in_k) \quad (3.7)$$

menghitung koreksi bobot (yang digunakan untuk memperbaiki  $w_{jk}$  nanti),

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (3.8)$$

dan mengirim  $\delta_k$  ke neuron lapis tersembunyi.

**Langkah 7:** Setiap neuron tersembunyi-2 ( $Z'_j, j=1\dots m$ ) jumlahkan bobot setiap neuron yang telah dikali dengan kesalahan informasinya,

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (3.9)$$

mengalikan dengan turunan fungsi aktivasinya untuk menghitung kesalahan informasinya,

$$\delta_j = \delta_{in_j} f'(z'_{in_j}) \quad (3.10)$$

menghitung koreksi bobotnya (yang digunakan untuk memperbaiki  $v_{ij}$  nanti)

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (3.11)$$

**Perbaiki bobot untuk proses selanjutnya**

**Langkah 8:** Setiap neuron keluaran ( $Y_k, k=1\dots o$ ) memperbaiki bobotnya ( $j=0\dots m$ )

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \quad (3.12)$$

Setiap neuron tersembunyi-1 ( $Z_j, j=1\dots p$ ) memperbaiki bobotnya

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \quad (2.13)$$

Setiap neuron tersembunyi-2 ( $Z'_j, j=1\dots m$ ) memperbaiki bobotnya

$$v'_{ij}(new) = v'_{ij}(old) + \Delta v_{ij} \quad (3.14)$$

**Langkah 9:** Lakukan pengujian kondisi henti.

Nilai *error* yang diizinkan dalam proses pelatihan dalam penelitian ini adalah dengan nilai galat sebesar 0.0001. Apabila nilai galat dari *error* lebih besar 0.0001 maka jaringan dianggap belum mengenali pola. Sementara nilai bilangan belajar yang diberikan adalah 0.3 dan 1. Nilai momentum adalah 0.8 sesuai dengan penelitian sebelum yang sudah dilakukan sebelumnya oleh Jiyo untuk memprediksi frekuensi maksimum lapisan F. Nilai ini menunjukkan kecepatan belajar dari suatu jaringan.

### 3.5 Metode Pelatihan Jaringan

Pelatihan dilakukan dengan menggunakan perangkat lunak (*software*) *MATLAB* versi 7.0.4 yang telah menyediakan fungsi-fungsi pelatihan dan pengujian pada JST dengan algoritma *backpropagation*.

Sebelum dilakukan pelatihan data input maupun target terlebih dahulu dilakukan normalisasi dengan bantuan rata-rata (mean) dan simpangan baku (standar deviasi). Untuk melakukan proses tersebut digunakan fungsi *prestd* sehingga data normal dengan mean=0 dan standar deviasi = 1. Fungsi *prestd* menggunakan *syntax*:

```
[pn,meanp,stdp,tn,ment,stdpt]=prestd(p,t)
```

Dengan  $p$  adalah matriks input dan  $t$  adalah matriks output

$p_n$  : matriks input yang ternormalisasi

$t_n$  : matriks target yang ternormalisasi.

$mean_p$  : mean pada matriks input asli ( $p$ ).

$std_p$  : standar deviasi pada matriks input asli ( $p$ ).

$mean_t$  : mean pada matriks target asli ( $t$ ).

$std_t$  : standar deviasi pada matriks input asli ( $t$ ).

Kemudian jaringan saraf tiruan dibangun dengan metode pelatihan gradient descent dengan momentum (*traingdm*) dengan *sintax*:

**`net = newff(PR,[ S1 S2...SN1],{TF1, TF2...TFN1},BTF,BLF,PF)`**

$PR$  : matriks berukuran  $R \times 2$  yang berisi nilai maksimum dan minimum, dengan  $R$  adalah jumlah variable input.

$S_i$  : jumlah neuron pada lapisan ke- $i$ , dengan  $i=1,2,3,\dots,N1$ .

$TF_i$  : fungsi aktivasi pada lapisan ke  $i$ , dengan  $i=1,2,3,\dots,N1$ .

$BTF$  : fungsi pelatihan jaringan.

$BLF$  : fungsi pelatihan untuk bobot.

$PF$  : fungsi kinerja.

Setelah jaringan saraf tiruan dibangun dengan metode *gradient descent* dengan momentum kemudian akan dilakukan identifikasi bobot dan bias dengan fungsi *net* yang memiliki persamaan sebagai berikut:

$$[W1,b1,W2,b2] = \text{net}(IW,b,LW)$$

sebelum dilakukan pelatihan, ada beberapa parameter pelatihan yang harus ditentukan yaitu: maksimum iterasi (*epoch*) dengan perintah *net.trainParam.epochs*, target gradient galat yang akan dicapai dengan perintah *net.trainParam.goal*, bilangan belajar dengan perintah *net.trainParam.lr*, selang tampilan tiap iterasi dengan perintah *net.trainParam.show*, dan momentum dengan perintah *net.trainParam.mc*.

Tahap selanjutnya adalah pelatihan jaringan. Pelatihan jaringan dilakukan dengan menggunakan fungsi *net* dengan persamaan sebagai berikut:

$$\text{net} = \text{train}(\text{net}, \text{pn}, \text{tn})$$

Proses pelatihan ini dilakukan secara berulang sampai dihasilkan suatu jaringan yang dapat memberikan suatu tanggapan yang benar terhadap nilai masukan. Tanggapan yang benar ini ditunjukkan oleh nilai *error* yang biasanya mempunyai nilai galat di bawah 0,0001. Apabila nilai RMS/ERROR menunjukkan di bawah 0,0001 maka jaringan sudah boleh dikatakan terlatih. Sedangkan jumlah iterasi maksimum sebesar 100.000 dengan selang tampilan per iterasi sebesar 200. Apabila pada pelatihan nilai epoch lebih dari 100.000 maka jaringan dianggap tidak konvergen, artinya JST tidak mampu melakukan pelatihan

pada sistem. Selanjutnya, pada pelatihan akan dicari parameter optimal dengan memvariasikan nilai momentum dan bilangan belajar.

*Source code* untuk pelatihan pada MATLAB terlihat pada tabel berikut

```

Clear;

% Data input & target

Data = [...]

P= Data ( input)

T= Data (target)

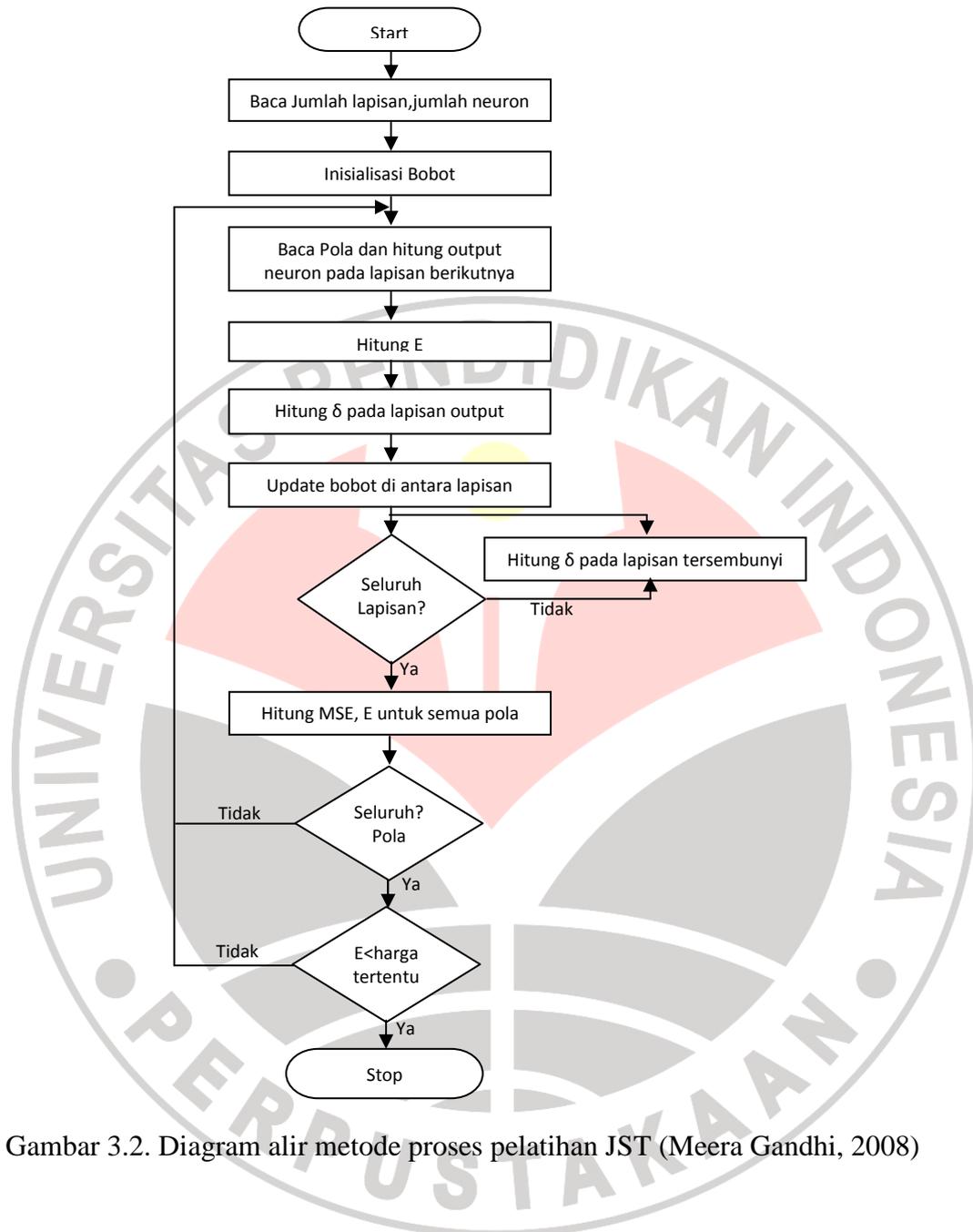
[pn,meanp,stdp,tn,meant,stdt] = prestd(P,T)

net = newff(PR,[ S1 S2...SN1},{TF1, TF2...TFN1},BTF,BLF,PF)

[WI,b1,W2,b2] = net (IW,b,LW)

```

Pelatihan JST dimulai dengan mempelajari pola pertama dengan umpan maju dan umpan balik kesalahannya, dilanjutkan dengan pola ke dua, kemudian pola ke tiga sampai pola seterusnya. Setelah ke semua pola tersebut dipelajari oleh JST, kemudian dihitung nilai *error*nya. Sampai proses ini, dikatakan bahwa JST telah melalui 1 *epoch*. Jika *error*nya lebih besar dari nilai yang ditetapkan atau jumlah epoch yang telah dilakukan kurang dari nilai yang telah ditetapkan, maka JST akan mempelajari kembali ke semua pola tersebut.



Gambar 3.2. Diagram alir metode proses pelatihan JST (Meera Gandhi, 2008)

### 3.6 Metode Pengujian Jaringan

Kemampuan jaringan untuk memprediksikan frekuensi maksimum lapisan E sporadis ditunjukkan dari nilai galat yang diperoleh dari hasil pelatihan jaringan semakin kecil galat maka jaringan dianggap semakin baik. Setelah pelatihan

sesuai dengan yang diharapkan maka tahap selanjutnya adalah tahap pengujian terhadap jaringan yang bertujuan untuk mengetahui apakah jaringan sudah mampu untuk memprediksikan frekuensi maksimum lapisan E sporadis.

Pengujian tersebut dilakukan dengan 2 tahap, yaitu pengujian terhadap data yang sudah dilatih dan pengujian terhadap data yang baru. Setelah bobot akhir dicapai maka jaringan akan diuji dengan menggunakan fungsi sebagai berikut:

$$[m,b,r] = \text{postreg}(a,t)$$

Dengan  $a$  adalah output jaringan syaraf tiruan, dan  $t$  adalah target.

$m$  : gradien garis hasil regresi linear. Jika output jaringan tepat sama dengan targetnya, maka gradien ini akan bernilai 1.

$b$  : titik perpotongan dengan sumbu  $y$ . Apabila output jaringan tepat sama dengan targetnya, maka perpotongan dengan sumbu nilai  $y$  ini akan bernilai 0

$r$  : koefisien korelasi antara output jaringan dengan target. Apabila output jaringan tepat sama dengan targetnya, maka koefisien korelasi ini akan bernilai 1

Untuk menguji kinerja model yang dikembangkan, dilakukan uji tingkat akurasi yang terdiri atas :

1. Tingkat akurasi internal yaitu tingkat keberhasilan JST dilihat dari nilai *error* terhadap jumlah seluruh data yang digunakan pada saat dilakukan pengujian.

2. Tingkat akurasi eksternal yaitu tingkat akurasi tiap kelas yang merupakan perbandingan antara pengenalan yang benar terhadap jumlah kelas yang terjadi.

Berdasarkan hasil uji tingkat akurasi, dapat didefinisikan kinerja model yang dikembangkan.

