

## **BAB III**

### **PREDIKSI CURAH HUJAN MENGGUNAKAN METODE JARINGAN SYARAF TIRUAN**

#### **3.1. Identifikasi Masalah**

Masalah yang diangkat pada penelitian ini adalah bagaimana memprediksi banyaknya curah hujan harian untuk tahun 2020 di Kota Pontianak. Curah hujan merupakan unsur iklim yang sangat penting bagi kehidupan di bumi. Besarnya curah hujan yang terjadi tidak dapat ditentukan secara pasti namun dapat diprediksi atau diperkirakan banyaknya curah hujan pada suatu periode tertentu. Salah satu metode yang dapat digunakan untuk memprediksi banyaknya curah hujan adalah menggunakan metode jaringan syaraf tiruan (JST). JST adalah sistem pengolahan informasi yang mempunyai karakteristik untuk kerja tertentu sebagaimana jaringan syaraf otak manusia yaitu mempelajari pola dan menghasilkan pola baru. JST mampu mempelajari data yang berbasis masa lampau lalu menghasilkan data berupa pola baru yang dapat digunakan di masa depan (Hermawan, 2006). Pada penelitian ini proses JST adalah mempelajari data historis besarnya curah hujan beberapa waktu yang lampau, lalu diramalkan berapa besarnya curah hujan yang terjadi akan pada masa yang akan datang.

Dalam penelitian ini penulis akan memprediksi curah hujan menggunakan faktor suhu rata-rata, kelembaban udara, kecepatan angin maksimum, dan arah angin saat kecepatan maksimum. Pada subbab selanjutnya akan dibahas bagaimana proses memprediksi banyaknya curah hujan menggunakan faktor-faktor tersebut dengan menggunakan JST.

#### **3.2. Klasifikasi Data**

Data yang digunakan pada penelitian ini adalah data yang diperoleh dari BMKG. Curah hujan akan diprediksi menggunakan metode jaringan syaraf tiruan. JST adalah metode yang memanfaatkan kemampuannya untuk belajar memahami suatu pola, oleh karena itu didalam JST terdapat algoritma pelatihan. Algoritma

Risyqaa Syafitri, 2020

*IMPLEMENTASI JARINGAN SYARAF TIRUAN PROPAGASI BALIK (JST-PB) UNTUK MEMPREDIKSI  
CURAH HUJAN DI KOTA PONTIANAK*

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

pelatihan ini bertujuan untuk mempelajari suatu data yang ada untuk menghasilkan suatu formula baru yang dimana hasilnya berupa nilai prediksi. Algoritma pelatihan ini terdiri dari data *input* dan data *target*. Data *input* akan berisi faktor-faktor yang berpengaruh terhadap curah hujan pada masa lampau. Semakin banyak data yang digunakan dalam pelatihan, maka jaringan akan mudah mengenali polanya dan diharapkan hasil prediksinya mendekati sempurna.

### 3.3 Jaringan Syaraf Tiruan-Propagasi Balik

JST adalah sistem pemrosesan informasi yang memiliki karakteristik mirip dengan syaraf biologis. JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi dengan asumsi bahwa (Siang, 2009):

- a) Pemrosesan informasi terjadi pada banyak elemen sederhana yaitu *neuron*.  
*Neuron* adalah unit pemroses informasi yang menjadi dasar pengoperasian pada JST. *Neuron* terdiri dari himpunan unit-unit yang dihubungkan dengan jalur koneksi. Jalur-jalur tersebut memiliki bobot/kekuatan yang berbeda-beda. Bobot yang bernilai positif akan memperkuat sinyal dan yang bernilai negatif akan memperlemah sinyal yang dibawanya. Jumlah struktur dan pola hubungan antar unit-unit tersebut akan menentukan arsitektur jaringan (dan juga model jaringan yang terbentuk). Suatu unit penjumlah yang akan menjumlahkan *input-input* sinyal yang sudah dikalikan dengan bobotnya. Pada penelitian  $x_1, x_2, \dots, x_n$  adalah unit-unit *input* dan  $w_1, w_2, \dots, w_p$  adalah bobot penghubung dari unit-unit tersebut ke unit keluaran  $y_m$ , maka unit penjumlahan akan memberikan keluaran sebesar  $\mu_j = x_1w_1 + x_2w_2 + \dots + x_nw_p$
- b) Sinyal dikirimkan diantara *neuron-neuron* melalui penghubung-penghubung.
- c) Penghubung antar *neuron* memiliki bobot yang akan memperkuat atau memperlemah sinyal.
- d) Untuk menentukan *output*, setiap *neuron* menggunakan fungsi aktivasi yang akan dikenakan pada jumlahan *input* yang diterima. Dalam jaringan syaraf tiruan, fungsi aktivasi dipakai untuk menentukan keluaran suatu neuron.

JST digunakan dalam beberapa aplikasi misalnya pengenalan pola (*pattern recognition*) seperti pengenalan pada suara atau tanda tangan, *signal processing*, dan peramalan. Pada penelitian ini aplikasi JST digunakan untuk peramalan

(*forecasting*), JST dapat dipakai untuk meramalkan apa yang akan terjadi di masa yang akan datang berdasarkan pola kejadian yang ada di masa yang lampau. Ini dapat dilakukan mengingat kemampuan jaringan syaraf tiruan untuk mempelajari dan membuat generalisasi dari apa yang sudah ada sebelumnya.

JST terdiri dari layer tunggal yang menjadikan JST memiliki keterbatasan dalam pengenalan pola. Kelemahan ini bisa ditanggulangi dengan menambahkan satu/beberapa layer tersembunyi diantara layer masukan dan keluaran. Meskipun penggunaan lebih dari satu layer tersembunyi memiliki kelebihan manfaat untuk beberapa kasus, tapi pelatihannya memerlukan waktu yang lama. Maka umumnya orang mulai mencoba dengan sebuah layer tersembunyi lebih dahulu.

Didalam JST terdapat penghubung antar *neuron* yang memiliki bobot untuk memperkuat atau memperlemah sinyal. Penghubung antar *neuron* ini berisi algoritma pelatihan. Algoritma pelatihan ini bertujuan untuk mempelajari suatu data yang ada untuk menghasilkan suatu formula baru yang hasilnya berupa prediksi.

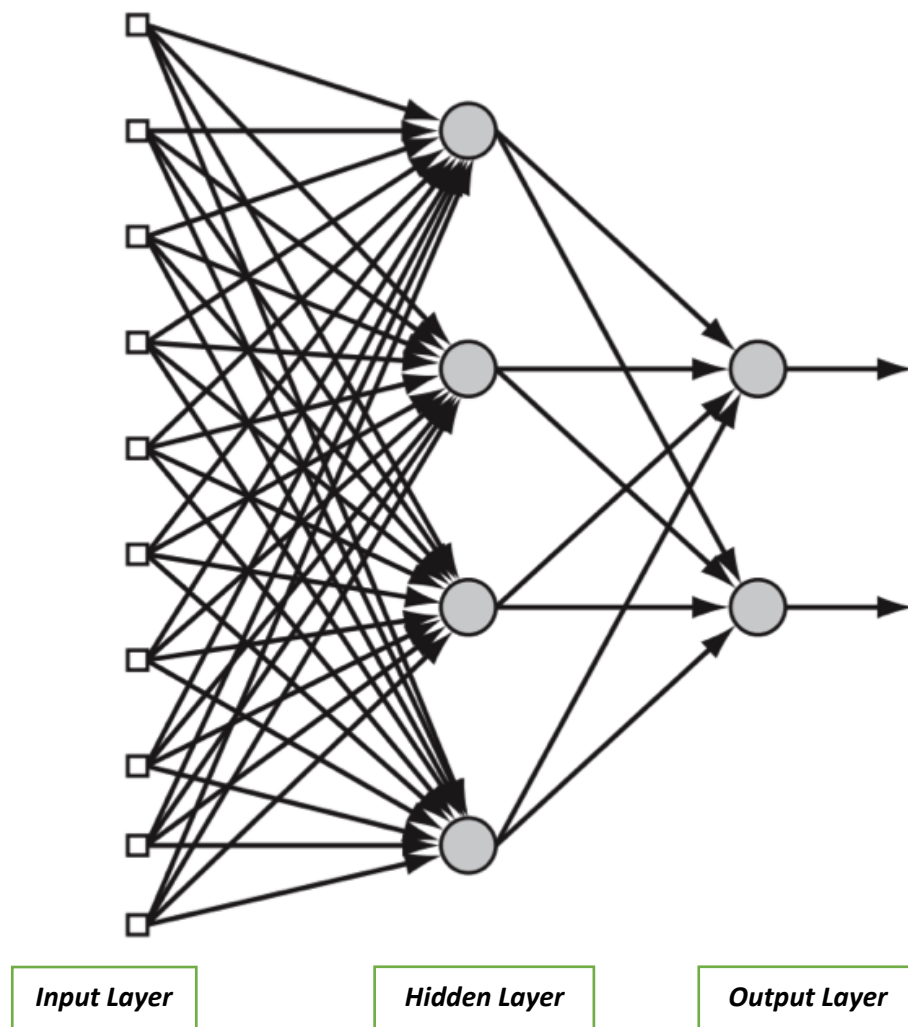
Berdasarkan cara menentukan bobot penghubung terdapat 2 macam algoritma pelatihan yaitu dengan supervise dan tanpa supervise. Pelatihan dengan supervise artinya terdapat sejumlah pasangan data (masukan-target keluaran) yang akan digunakan untuk melatih jaringan hingga diperoleh hasil yang diinginkan. Pasangan data tersebut berfungsi untuk melatih jaringan hingga diperoleh bentuk yang terbaik. Pada setiap kali pelatihan, suatu input diberikan ke jaringan. Jaringan akan memproses dan mengeluarkan keluaran. Selisih antara keluaran jaringan dengan target yang dimiliki merupakan kesalahan yang terjadi. Jaringan akan memodifikasi bobot sesuai dengan kesalahan tersebut. Sebaliknya, dalam pelatihan tanpa supervisi (*unsupervised learning*) tidak ada data pasangan yang akan mengarahkan proses pelatihan.

Jaringan propagasi balik merupakan salah satu model yang menggunakan pelatihan dengan supervisi. Propagasi balik melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang akan digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan dengan pola yang dipakai selama pelatihan.

Jaringan syaraf tiruan ditentukan oleh tiga tahapan:

- a. Pola hubungan antar *neuron* atau yang disebut arsitektur jaringan  
Pada penelitian ini digunakan arsitektur propagasi balik yang dimana memiliki beberapa unit yang ada dalam satu atau lebih layer tersembunyi. Arsitektur jaringan propagasi balik dapat dilihat di Gambar 3.1.
- b. Metode untuk menentukan bobot penghubung.  
Pada penelitian ini digunakan jaringan pelatihan algoritma propagasi balik (*backpropagation*). Pelatihan propagasi balik meliputi tiga fase, yaitu fase maju, fase balik, dan fase modifikasi bobot.

**Fase pertama** adalah fase maju. Pola masukan dihitung maju mulai dari layer masukan hingga layer keluaran menggunakan fungsi aktivasi yang ditentukan. Selama propagasi maju, sinyal masukan dipropagasikan ke layer tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit layer tersembunyi akan dilanjutkan dipropagasikan maju lagi ke layer tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan. Berikutnya, keluaran jaringan dibandingkan dengan target yang harus dicapai. Selisih diantaranya adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.



Gambar 3. 1. Arsitektur Jaringan Syaraf Tiruan

**Fase kedua** adalah fase balik. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran. Berdasarkan kesalahannya, dihitung faktor yang dipakai untuk mendistribusikan kesalahan di unit keluaran ke semua unit tersembunyi yang terhubung langsung dengan keluaran. Faktor juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran. Dengan cara yang sama, dihitung faktor di setiap unit di layer tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga semua faktor di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

**Fase ketiga** adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi. Setelah semua faktor dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor neuron di layar atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke layar keluaran didasarkan atas yang ada di unit keluaran. Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan.

c. Fungsi aktivasi.

Dalam jaringan syaraf tiruan, fungsi aktivasi dipakai untuk menentukan keluaran suatu neuron. Argumen fungsi aktivasi adalah net masukan. Jika  $net = \sum x_n w_p$ , maka fungsi aktivasinya adalah  $f(net) = f(\sum x_n w_p)$ . Fungsi yang dipakai pada pelatihan propagasi balik harus memenuhi beberapa syarat yaitu: kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Pada penelitian ini fungsi yang dapat memenuhi ketiga syarat tersebut adalah fungsi sigmoid biner yang bernilai 0 sampai 1. Fungsi sigmoid biner dapat ditulis:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Fungsi sigmoid sering dipakai karena nilai fungsinya yang terletak antara 0 dan 1 dan dapat diturunkan dengan mudah.

$$f'(x) = f(x) (1 - f(x))$$

Selanjutnya akan dibahas tahapan kerja Jaringan Syaraf Tiruan-Propagasi Balik untuk memprediksi banyaknya curah hujan. Tahapan tersebut adalah sebagai berikut:

1. Mempersiapkan Data

Langkah pertama yang dilakukan adalah mempersiapkan data, data didapatkan dari situs BMKG. Data yang didapatkan berbentuk tabel, data tersebut berisi beberapa parameter yaitu suhu rata-rata, kelembaban udara, kecepatan angin maksimum, angin saat kecepatan maksimum dan curah hujan. .

Data tersebut harus dibentuk kedalam pola yang nantinya akan dipelajari oleh JST. Pola tersebut berisi data masukan, yaitu faktor yang mempengaruhi banyak curah hujan dan data target keluaran (curah hujan) sebagai pasangan data. Penentuan pola bertujuan untuk membagi data antara data pelatihan, yaitu data yang digunakan untuk mendapatkan bobot yang optimal, akan tetapi kriteria penghentian didasarkan atas data pengujian dan data pengujian yaitu data yang digunakan untuk mendapatkan keseimbangan antara pengenalan pola pelatihan secara benar dan respon yang baik untuk pola lain yang sejenis.

## 2. Menentukan Arsitektur Jaringan

Dalam arsitektur jaringan syaraf tiruan terdiri dari tiga lapisan, yaitu lapisan *input*, lapisan *hidden* dan lapisan *output*. Lapisan *input* terdiri dari faktor-faktor yang berpengaruh pada curah hujan. Pada lapisan *hidden* terdiri dari sejumlah *node* yang akan diuji untuk mengetahui jumlah *node* yang optimal untuk dilakukan pengujian. Pada lapisan *output* akan mengeluarkan data curah hujan dalam satuan mm.

## 3. Normalisasi Data

Setelah terbentuk pola data selanjutnya akan dilakukan normalisasi data. Normalisasi data ini bertujuan agar data tidak memiliki nilai dengan jarak yang jauh berbeda serta lebih mudah dalam melakukan pelatihan. Pada algoritma jaringan syaraf tiruan propagasi balik ini digunakan fungsi aktivasi sigmoid biner dengan nilai range 0 sampai 1. Sebelum melakukan pelatihan data perlu dinormalisasi terlebih dahulu, salah satu contohnya kedalam range 0,1 sampai 0,9 menggunakan persamaan berikut:  $x' = 0,8(x - b)(a - b) + 0,1$

dimana:

$x'$ : data hasil normalisasi

$x$ : data asli/data awal

$a$ : nilai maksimum data asli

$b$ : nilai minimum data asli

## 4. Pemilihan Algoritma Pelatihan Jaringan

Algoritma pelatihan jaringan yang digunakan adalah algoritma jaringan propagasi balik yang memiliki tiga lapisan dengan fungsi aktivasi sigmoid biner. Masalah yang dihadapi dalam algoritma propagasi balik adalah lamanya iterasi

yang harus dilakukan. Propagasi balik tidak dapat memberikan kepastian tentang berapa iterasi (*epoch*) yang harus dilalui untuk mencapai kondisi yang diinginkan. Oleh karena itu peneliti akan melakukan pengujian untuk menentukan parameter-parameter jaringan dibuat sehingga menghasilkan keluaran sesuai keinginan. Parameter-parameter lainnya yang harus ditentukan adalah momentum, laju pemahaman dan batas galat.

Pada standar propagasi balik, perubahan bobot didasarkan atas gradien yang terjadi untuk pola yang dimasukkan saat itu. Gradien adalah kemiringan suatu garis, dimana didalam pelatihan yaitu pola yang dimasukkan adalah perbandingan antara nilai masukan dan nilai target. Modifikasi yang dapat dilakukan adalah melakukan perubahan bobot yang didasarkan atas arah gradien pola terakhir dan pola sebelumnya yang disebut momentum. Jadi tidak hanya pola masukan terakhir saja yang diperhitungkan. Penambahan momentum dimaksudkan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan yang lain. Apabila beberapa data terakhir yang diberikan ke jaringan memiliki pola serupa yang berarti arah gradien sudah benar, maka perubahan bobot dilakukan secara cepat. Namun apabila data terakhir yang dimasukkan memiliki pola yang berbeda dengan pola sebelumnya, maka perubahan dilakukan secara lambat.

Laju pemahaman ( $\alpha$ ) merupakan suatu konstanta yang dipakai dalam seluruh iterasinya. Perubahan dapat dilakukan dengan memberikan laju pemahaman yang berbeda-beda untuk setiap bobotnya atau bahkan laju pemahaman yang berbeda-beda untuk tiap bobot dalam tiap iterasinya. Apabila perubahan bobot berada dalam arah yang sama dalam beberapa pola terakhir, maka laju pemahaman yang harus ditambah. Sebaliknya apabila arah perubahan bobot dua pola terakhir berbeda maka laju pemahaman untuk bobot tersebut harus dikurangi.

Batas galat atau batas kesalahan adalah suatu statistik yang menunjukkan besarnya galat atau kesalahan yang dapat diterima atas suatu nilai-duga (*estimate*) sebagai konsekuensi dari ukuran acak yang diambil dalam suatu survei. Pada penelitian ini, pola masukan dirambatkan melalui lapisan masukan sampai lapisan keluaran, lalu akan dicocokkan dengan keluaran target awal. Jika



terdapat perbedaan nilai diantara keduanya, selisih keduanya disebut dengan galat dan galat tersebut akan ditentukan, sehingga sebelum nilai galat sesuai dengan yang diinginkan maka pelatihan akan terus dilakukan. Nilai parameter galat berpengaruh terhadap hasil akurasi data uji karena jika pada saat pelatihan jaringan nilai parameter galat tidak terpenuhi maka hasil pengujian terhadap data uji akan menghasilkan hasil akurasi yang rendah. Semakin kecil galat yang akan ditentukan maka hasil tingkat akurasi yang lebih tinggi.

Tahapan awal algoritma propagasi balik adalah menetapkan nilai parameter berikut:

- i. Inisialisasi bobot dan bias secara acak dengan bilangan dari 0 sampai 1 setelah dilakukan normalisasi data.
- ii. Tetapkan jumlah *epoch* maksimum, laju pembelajaran, momentum dan batas galat.

Algoritma di atas memiliki dua kondisi pemberhentian yaitu *epoch* yang telah mencapai *epoch* maksimum atau  $MSE < \text{batas galat}$ . MSE (*Mean Squared Error*) adalah metode untuk mengevaluasi kesalahan didalam peramalan. Masing-masing kesalahan dikuadratkan. Metode ini menghasilkan kesalahan-kesalahan yang kecil untuk hasil yang lebih baik. Selengkapnya algoritma pelatihan propagasi balik untuk jaringan dengan satu lapisan tersembunyi menggunakan fungsi aktivasi *sigmoid biner* adalah sebagai berikut (Siang, 2009):

Langkah 0: Inisialisasi parameter.

Langkah 1: ketika kondisi berhenti salah, lakukan langkah 2-9.

Langkah 2: untuk masing-masing pasangan data atau disebut pasangan *training*, lakukan langkah 3-8.

**Fase I : Propagasi maju (*Feedforward*):**

Langkah 3: Tiap-tiap unit *input* ( $x_i, i = 1,2,3, \dots n$ ) menerima sinyal  $x_i$  dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya yaitu lapisan tersembunyi.

Langkah 4: Tiap-tiap unit tersembunyi ( $z_j, j = 1,2,3, \dots p$ ) menjumlahkan sinyal-sinyal *input* terbobot :  $z_{net_j} = v_{j_0} + \sum_{i=1}^n x_i v_{ji}$  gunakan fungsi

aktivasi untuk menghitung sinyal *output*nya :  $z_j = f(z_{net_j}) = \frac{1}{1+e^{-z_{net_j}}}$ , lalu kirim sinyal ini ke semua unit pada layer di atasnya yaitu unit keluaran.

Langkah 5: Tiap-tiap unit *output* ( $y_k, k = 1, 2, 3, \dots, m$ ) menjumlahkan sinyal-sinyal *input* terbobot.  $Y_{net_k} = w_{k_0} + \sum_{i=1}^p Z_j w_{kj}$  gunakan fungsi aktivasi untuk menghitung sinyal *output*nya :  $Y_k = f(Y_{net_k}) = \frac{1}{1+e^{-y_{net_k}}}$ . Dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya yaitu unit-unit *output*.

### Fase II : Propagasi balik (*Backpropagation*):

Langkah 6: Hitung informasi *error*-nya:  $\delta_k = (t_k - y_k)f'(y_{net_k})$  berdasarkan kesalahan di setiap unit *output*nya. Kemudian hitung koreksi bobot yang akan digunakan untuk memperbaiki nilai  $w_{kj}$  dengan laju pemahaman  $\alpha$ , yaitu

$$\Delta w_{kj} = \alpha \delta_k z_j ; k = 1, 2, \dots, m ; j = 1, 2, \dots, p$$

Langkah 7: Hitung *error*-nya di unit tersembunyi ( $Z_j, j = 1, 2, 3, \dots, p$ ) berdasarkan kesalahan di setiap unit tersembunyi:

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{kj}$$

Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error*:

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

Kemudian hitung koreksi bobot yang nantinya akan digunakan untuk memperbaiki nilai  $v_{ji}$ :

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

### Fase III : Modifikasi bobot

Langkah 8: Hitung semua perubahan bobot. Tiap-tiap unit *output* ( $k = 1, 2, 3, \dots, m$ ) memperbaiki bias dan bobotnya ( $j = 1, 2, 3, \dots, p$ ) :

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

Tiap-tiap unit tersembunyi ( $j = 1, 2, 3, \dots, p$ ) memperbaiki bias dan bobotnya ( $i = 0, 1, 2, \dots, n$ ) :

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

Langkah 9: Tes kondisi berhenti. Selama kondisi berhenti tidak terpenuhi lakukan langkah ke-2 sampai ke-8. Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola dengan menggunakan fase propagasi maju saja (langkah 3 dan 4) untuk menentukan keluaran jaringan.

Keterangan:

$X_i$ : Unit input ke  $i$

$Z_j$ : Hidden unit ke  $j$

$Y_k$ : Unit output ke  $k$

$T_k$ : Pola target ke  $k$

$V_{0j}$ : Bias untuk hidden unit ke  $j$

$V_{ij}$ : Bobot antara unit input ke  $i$  dengan hidden unit ke  $j$

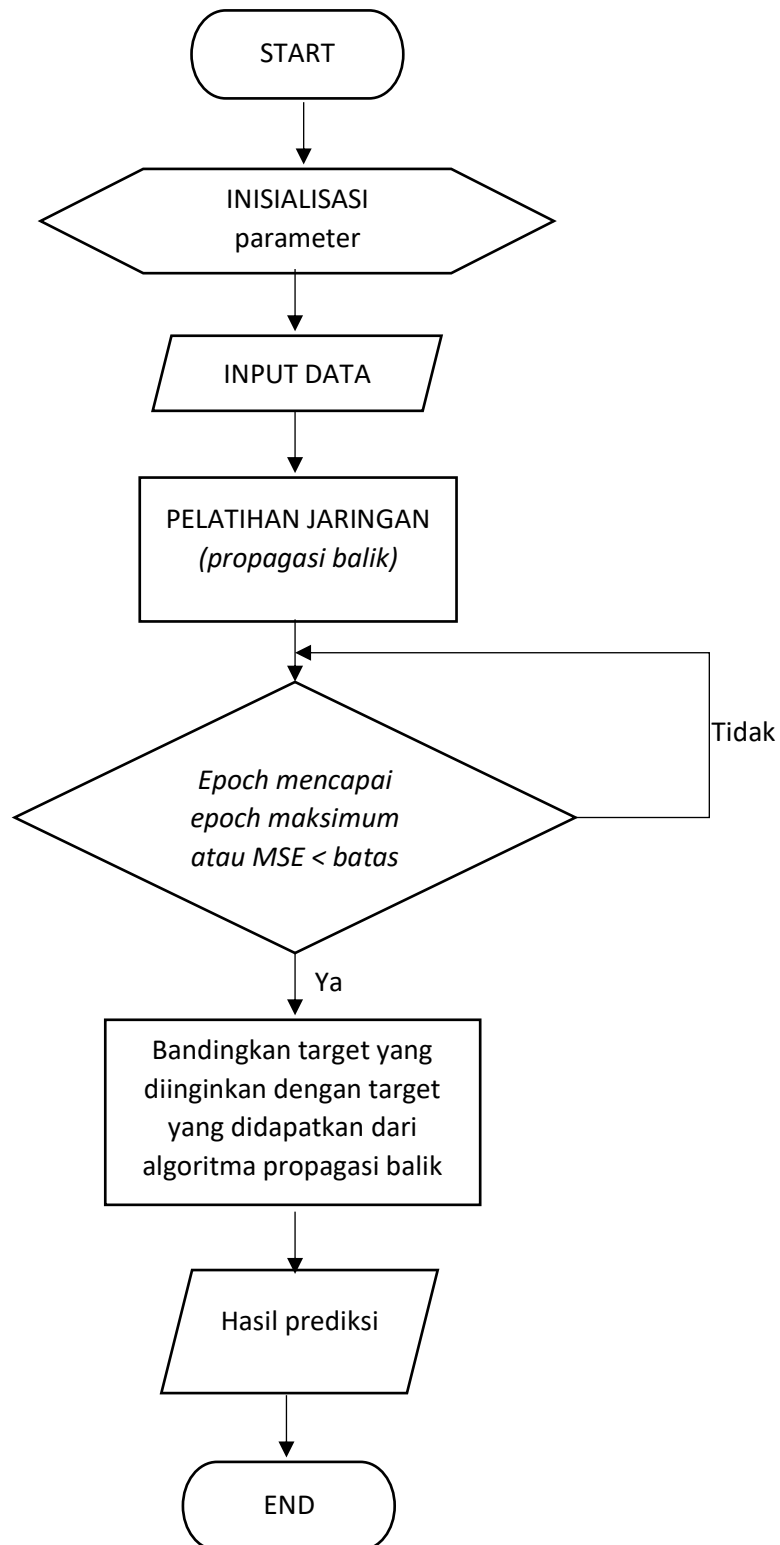
$W_{0k}$ : Bias untuk unit output ke  $k$

$W_{jk}$ : Bobot antara hidden unit ke  $j$  dengan unit output ke  $k$

$\delta_k$ : error

## 5. Analisis Prediksi

Hasil dari pengujian ialah tingkat keakuratan antara nilai dugaan ( $y$ -prediksi) dengan  $y$ -aktual berdasarkan dua parameter, yaitu  $R$  dan  $MSE$ . Nilai akurasi semakin baik ketika  $R$  mendekati 1 dan akurasi terbaik diambil pada nilai  $MSE$  yang terkecil atau yang mendekati 0. Untuk memastikan hasil prediksi, nilai prediksi yang didapatkan akan dibandingkan dengan nilai aktual. Cara kerja jaringan syaraf tiruan propagasi balik dapat dilihat di Gambar 3.2



Gambar 3. 2. Flowcart Jaringan Syaraf Tiruan Propagasi Balik untuk Memprediksi Curah Hujan