

## PENGEMBANGAN MODEL JARINGAN SYARAF TIRUAN UNTUK PRAKIRAAN TINGGI MUKA AIR HARIAN SUNGAI BENGAWAN SOLO STASIUN JURUG

Mina Pusporani<sup>1</sup>  
Palgunadi<sup>2</sup>  
Sri Huning A<sup>3</sup>

[mail@usahidsolo.ac.id](mailto:mail@usahidsolo.ac.id)

### ABSTRAKSI

Bengawan Solo mempunyai Daerah Aliran Sungai (DAS) seluas 16.100 km<sup>2</sup>- merupakan sungai yang terpanjang di Pulau Jawa sehingga membutuhkan suatu sistem pengelolaan yang baik. Manajemen sungai merupakan salah satu bidang hidrologi operasional yang cukup menantang. Prakiraan aliran sungai dibutuhkan sebagai informasi dasar pada berbagai masalah yang berhubungan dengan desain dan operasi sistem-sistem sungai. Jaringan syaraf tiruan (JST) telah dikenal dengan kemampuannya dalam mengidentifikasi fungsi-fungsi non-linear, antara lain proses hidrologi. Dalam penelitian ini, model jaringan backpropagation digunakan untuk memprakirakan tinggi muka air (tma) harian Sungai Bengawan Solo stasiun Jurug. Data tinggi muka air harian dan curah hujan harian digunakan sebagai data pelatihan dan pengujian. Untuk menguji performa jaringan, dalam penelitian dibuat tiga model JST. Ketiga model JST – BP1, BP2, dan BP3 – merupakan multilayer perceptron dengan satu hidden layer. BP1 memiliki input berupa tma (m) dan curah hujan (mm) empat hari sebelum prakiraan dan output berupa prakiraan tma. BP2 dengan input dan output yang sama hanya berbeda dalam satuan – tma (dm) dan curah hujan (cm). BP3 dengan input berupa selisih tma (dm) dan curah hujan (cm) serta output berupa prakiraan selisih tma. Ketiga model dilatih menggunakan algoritma Levenberg-Marquardt dan jumlah hidden neuron yang beragam [3..10]. Nilai mse dan korelasi setiap jaringan dibandingkan untuk mendapatkan model dan parameter jaringan yang optimal. Dari hasil penelitian didapat kesimpulan bahwa model jaringan yang paling optimal dalam kasus ini adalah model BP3 dengan satu hidden layer dan empat hidden neuron karena menghasilkan nilai error terkecil dibandingkan dengan model yang lain.

Kata Kunci : Backpropagation, jaringan syaraf tiruan, Perceptron.

### I. PENDAHULUAN

Bengawan Solo merupakan sungai yang terpanjang di Pulau Jawa, yang meliputi Provinsi Jawa Tengah dan Jawa Timur. Berdasarkan geografi, Wilayah Sungai Bengawan Solo

1. **Dosen Jurusan Teknik Informatika, Universitas Sahid Surakarta**  
Jl. Adi Sucipto 154 Surakarta
2. **Dosen Jurusan Teknik Informatika, Universitas Sahid Surakarta**  
Jl. Adi Sucipto 154 Surakarta
3. **Dosen Jurusan Teknik Informatika, Universitas Sahid Surakarta**  
Jl. Adi Sucipto 154 Surakarta

dibagi menjadi Sub Satuan Wilayah Sungai (Sub SWS) Bengawan Solo Hulu dan Sub SWS Bengawan Solo Hilir. Pemanfaatan air Bengawan Solo antara lain untuk irigasi, kebutuhan rumah tangga melalui PDAM dan non-PDAM, industri, dan pertanian.

Prakiraan aliran sungai dibutuhkan sebagai informasi dasar pada berbagai masalah yang berhubungan dengan desain dan operasi sistem-sistem sungai. Dalam hal ini, hubungan antara curah hujan (*rainfall*) dan aliran permukaan (*runoff*) telah dieksploitasi secara luas pada banyak model *rainfall-runoff* konseptual (Danh, 1999). Proses transformasi *rainfall-runoff* telah diketahui sebagai proses yang sangat kompleks dan non-linear (Alvisi, 2006).

Jaringan syaraf tiruan (JST) bisa jadi adalah teknik yang paling berhasil saat ini dengan struktur matematis fleksibel yang mampu mengidentifikasi hubungan non-linear kompleks antara data input dan output (Dibike, 2000). Alat pemodelan empiris ini didesain untuk mengemulasi kemampuan manusia dalam pengenalan pola melalui pemrosesan input multipel secara paralel (Aqil, 2006). Teknologi JST telah membuktikan banyak hasil menjanjikan dalam pemodelan sistem non-linear yang kompleks, dan aplikasinya dalam bidang hidrologi dan sumber daya air telah banyak dilaporkan, seperti prakiraan aliran sungai, prakiraan tinggi muka air, simulasi *rainfall-runoff*, dan prakiraan cuaca (Aqil, 2006).

Dalam penelitian ini, akan dikembangkan model JST untuk prakiraan tinggi muka air harian berdasarkan curah hujan dan tinggi muka air pada Sungai Bengawan Solo stasiun Jurug. Penelitian ini bertujuan untuk mengetahui model jaringan JST *backpropagation* yang paling optimal dan pengaruh variabel-variabel (jumlah *hidden neuron*, parameter input, dan satuan tma) terhadap performansi JST. Hasil penelitian ini diharapkan dapat membantu Perum Jasa Tirta I - sebagai pihak pengelola Sungai Bengawan Solo- dalam menentukan kebijakan langkah berkaitan dengan manajemen sungai.

## II. JARINGAN SYARAF TI RUAN

### 2.1. Definisi JST

Jaringan Syaraf Tiruan adalah paradigma pemrosesan suatu informasi yang terinspirasi oleh sistem sel syaraf biologi, sama seperti otak yang memproses suatu informasi. Elemen mendasar dari paradigma tersebut adalah struktur yang baru dari sistem pemrosesan informasi. Jaringan Syaraf Tiruan, seperti manusia, belajar dari suatu contoh. Jaringan Syaraf Tiruan dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran. Sistem jaringan syaraf tiruan merupakan suatu sistem yang memiliki pengetahuan dalam menganalisa suatu masalah dan melakukan pekerjaan-pekerjaan klasifikasi pola, pemodelan sistem dan memori asosiasi. Klasifikasi pola digunakan untuk menganalisis pola-pola masukkan

dengan cara mencari kemiripan pola-pola masukan. Pemodelan sistem digunakan untuk pembuat simulasi sistem yang mampu menghasilkan keluaran dari suatu pola masukan yang akan disimulasikan. Sedangkan memori asosiasi digunakan untuk menganalisis pola masukan yang tidak lengkap, misalnya pola masukan memiliki derau, terpotong-potong, rusak dan hanya bisa tampil sebagian (Yani, 2005).

## 2.2. Model Dasar JST

Jaringan syaraf tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi. JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa (1) Pemrosesan informasi terjadi pada banyak elemen sederhana (*neuron*), (2) Sinyal dikirimkan di antara *neuron-neuron* melalui penghubung-penghubung, dan (3) Penghubung antar *neuron* memiliki bobot yang akan memperkuat atau memperlemah sinyal.

Untuk menentukan output, setiap *neuron* menggunakan fungsi aktivasi (biasanya bukan fungsi linier) yang dikenakan pada jumlahan input yang diterima. Besarnya output ini selanjutnya dibandingkan dengan suatu ambang batas. JST ditentukan oleh tiga hal yaitu (1) Pola hubungan antar *neuron* (disebut arsitektur jaringan), (2) Metode untuk menentukan bobot penghubung (disebut metode *training/learning/ algoritma*), dan (3) Fungsi aktivasi.

*Neuron* adalah unit pemroses informasi yang menjadi dasar dalam pengoperasian jaringan syaraf tiruan. *Neuron* terdiri atas tiga elemen pembentuk :

1. 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 negatif akan memperlemah sinyal yang dibawahnya. Jumlah, struktur, dan pola hubungan antar unit-unit tersebut akan menentukan arsitektur jaringan (dan juga model jaringan yang akan terbentuk).
2. Suatu unit penjumlah yang akan menjumlahkan input-input sinyal yang sudah dikalikan dengan bobotnya.
3. Fungsi aktivasi yang akan menentukan apakah sinyal dari *input neuron* akan diteruskan ke *neuron* lain ataukah tidak (Siang, 2005).

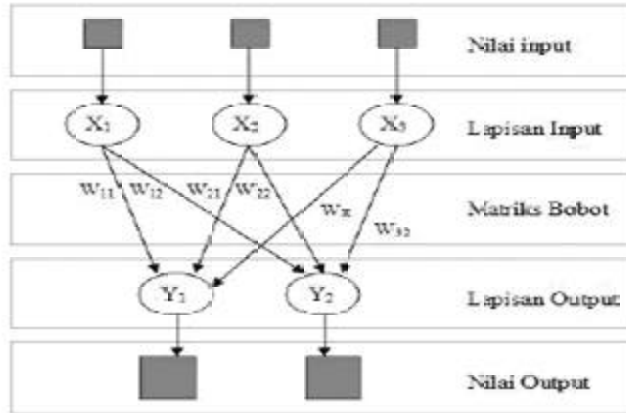
## 2.3. Arsitektur Jaringan

Beberapa arsitektur jaringan yang sering dipakai dalam jaringan syaraf tiruan antara lain :

1. Jaringan layar tunggal (*single layer network*)

Dalam jaringan ini, sekumpulan *input neuron* dihubungkan langsung dengan sekumpulan

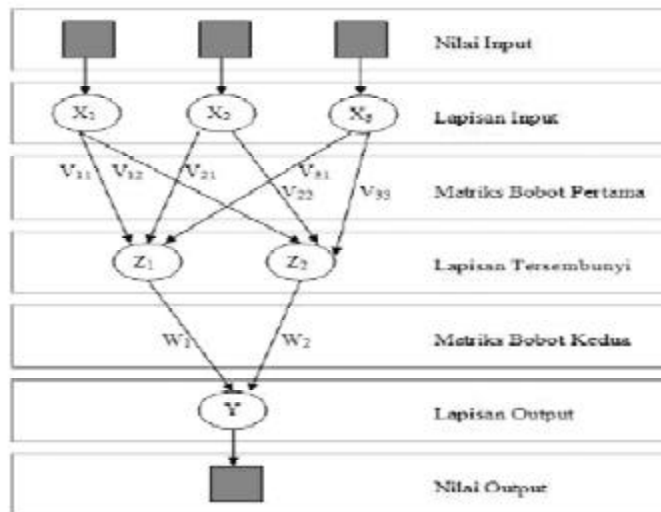
outputnya. Dalam beberapa model (misal *perceptron*), hanya ada sebuah unit *neuron* output.



Gambar 1. Jaringan layar tunggal

## 2. Jaringan layar jamak (*multilayer network*)

Dalam jaringan ini, selain unit input dan output, ada unit-unit lain (sering disebut layar tersembunyi). Dimungkinkan pula ada beberapa layar tersembunyi. Sama seperti pada unit input dan output, unit-unit dalam satu layar tidak saling berhubungan.



Gambar 2 . Jaringan layar jamak

Jaringan layar jamak dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan layar tunggal, meskipun kadangkala proses pelatihan lebih kompleks dan lama.

### 3. Jaringan *reccurent*

Model jaringan *reccurent* mirip dengan jaringan layar tunggal ataupun ganda. Hanya saja, ada *neuron* output yang memberikan sinyal pada unit input (sering disebut *feed-back loop*).

## 2.4. Fungsi Aktivasi

Dalam jaringan syaraf tiruan, fungsi aktivasi dipakai untuk menentukan keluaran suatu *neuron*. Argumen fungsi aktivasi adalah net masukan (kombinasi linier masukan dan bobotnya).

Beberapa fungsi aktivasi yang sering dipakai adalah sebagai berikut :

#### 1. Fungsi *threshold* (ambang batas)

$$Y = \begin{cases} 0, & \text{jika } x < \theta \\ 1, & \text{jika } x \geq \theta \end{cases}$$

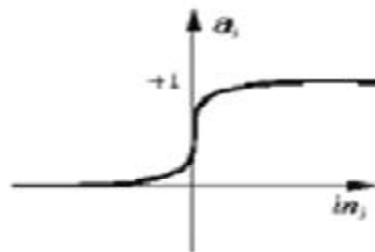
Untuk beberapa kasus, fungsi *threshold* yang dibuat tidak berharga 0 atau 1, tapi berharga -1 atau 1 (sering disebut *threshold* bipolar)

$$Y = \begin{cases} 1, & \text{jika } x \geq \theta \\ -1, & \text{jika } x < \theta \end{cases}$$

#### 2. Fungsi sigmoid

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

(1)



Gambar 3. Fungsi sigmoid

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

$$f'(x) = f(x) (1 - f(x)) \quad (2)$$

### 3. Fungsi identitas

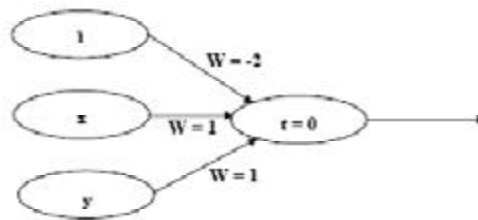
$$f(x) = x \quad (3)$$

Fungsi identitas sering dipakai apabila menginginkan keluaran jaringan berupa sembarang bilangan riil (bukan hanya pada range  $[0, 1]$  atau  $[-1, 1]$ ).

### 2.5. Bias

Kadang-kadang dalam jaringan ditambahkan sebuah unit masukan yang nilainya selalu = 1. Unit yang sedemikian itu disebut bias. Bias dapat dipandang sebagai sebuah input yang nilainya = 1.

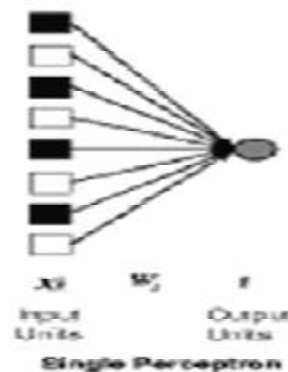
$$y_i = wx_i + b \quad (4)$$



Gambar 4. Jaringan dengan satu bias pada *input layer*

### 2.6. Perception

*Perceptron* terdiri atas beberapa unit masukan (ditambah sebuah bias), dan memiliki sebuah unit keluaran. Hanya saja fungsi aktivasi bukan merupakan fungsi biner (atau bipolar), tetapi memiliki kemungkinan nilai -1, 0, atau 1.



Gambar 5. *Single perceptron*

Untuk sebuah harga *threshold*  $\epsilon$  yang ditentukan :

$$f(\text{net}) = \begin{cases} 1 & \text{jika } \text{net} > \epsilon \\ 0 & \text{jika } -\epsilon \leq \text{net} \leq \epsilon \\ -1 & \text{jika } \text{net} < -\epsilon \end{cases}$$

secara geometris, fungsi aktivasi membentuk dua garis sekaligus, masing-masing dengan persamaan :

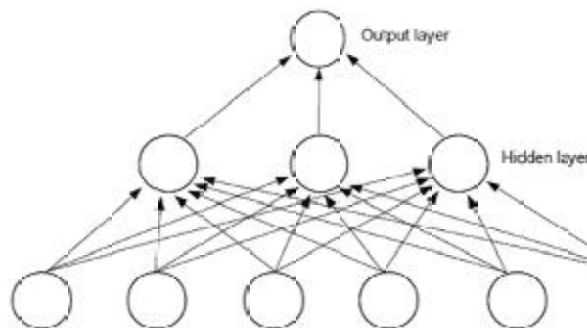
$$w_1x_1 + w_2x_2 + \dots + w_nx_n + b = \epsilon \text{ dan} \quad (5)$$

$$w_1x_1 + w_2x_2 + \dots + w_nx_n + b = -\epsilon \quad (6)$$

## 2.7. Backpropagation

*Backpropagation* adalah sebuah metode pelatihan (kalibrasi) jaringan syaraf multilayer yang sistematis. *Backpropagation* menggunakan satu set pasangan nilai input dan output (disebut pola). Sebuah pola input diumpankan ke jaringan untuk menghasilkan sebuah output, yang kemudian dibandingkan dengan pola output aktual. Jika tidak ada perbedaan antara output jaringan dan output aktual, maka pembelajaran tidak diperlukan. Jika sebaliknya, bobot – yang mengekspresikan kontribusi *input neuron* kepada *hidden neuron*, dan *hidden neuron* kepada output – diubah (dengan arah terbalik dari *output layer* ke *input layer*). Karena pelatihan tersebut menggunakan output aktual, metode *Backpropagation* disebut juga metode pelatihan dengan supervisi (Danh, 1999).

Algoritma *Backpropagation* terdiri atas tiga fase. Fase pertama adalah fase maju, yaitu pola input dihitung maju mulai dari lapisan input hingga lapisan output menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur, yaitu berdasarkan selisih antara output jaringan dengan target yang diinginkan dipropagasikan mundur mulai dari garis yang berhubungan langsung dengan lapisan output. Fase ketiga adalah modifikasi bobot untuk menurunkan *error* yang terjadi.



Gambar 6. JST *backpropagation* dengan satu *hidden layer*

Proses matematis pada fase maju :

1. Masing-masing unit masukan ( $X_i$ ,  $i = 1, \dots, n$ ) menerima sinyal masukan  $X_i$  dan sinyal tersebut disebarkan ke unit-unit berikutnya (unit-unit lapisan tersembunyi)
2. Masing-masing unit di lapisan tersembunyi dikalikan dengan bobot dan dijumlahkan serta ditambah dengan biasnya:

$$z_{in} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (7)$$

3. Kemudian menghitung sesuai dengan fungsi aktivasi yang digunakan:

$$z_j = f(z_{in_j}) \quad (8)$$

Bila yang digunakan adalah fungsi sigmoid, maka bentuk fungsi tersebut adalah:

$$Z_j = \frac{1}{1 + \exp(-z_{in_j})} \quad (9)$$

Kemudian mengirim sinyal tersebut ke semua unit keluaran

4. Masing-masing unit keluaran ( $y_k$ ,  $k = 1, 2, 3, \dots, m$ ) dikalikan dengan bobot dan dijumlahkan:

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (10)$$

Menghitung kembali sesuai dengan fungsi aktivasi

$$y_k = f(y_{in_k}) \quad (11)$$

Fase mundur :

1. Masing-masing unit keluaran ( $Y_k$ ,  $k = 1, \dots, m$ ) menerima pola target sesuai dengan pola masukan saat pelatihan dan dihitung galatnya:

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (12)$$

Karena  $f'(y_{in_k}) = y_k (1 - y_k)$  menggunakan fungsi sigmoid, maka:

$$\begin{aligned} f'(y_{in_k}) &= f(y_{in_k}) (1 - f(y_{in_k})) \\ &= y_k (1 - y_k) \end{aligned} \quad (13)$$

Menghitung perbaikan bobot (kemudian untuk memperbaiki  $w_{jk}$ ).

$$\Delta W_{kj} = \alpha \cdot \delta_k \cdot Z_j \quad (14)$$



Menghitung perbaikan bias:

$$\Delta W_{0,k} = \alpha \cdot \delta_k \quad (15)$$

Dan menggunakan nilai  $\delta_k$  pada semua unit lapisan sebelumnya (*hidden layer*).

2. Masing-masing bobot yang menghubungkan unit-unit lapisan keluaran dengan unit-unit pada lapisan tersembunyi ( $Z_j$ ,  $j = 1, \dots, p$ ) dikalikan delta dan dijumlahkan sebagai masukan ke unit-unit lapisan berikutnya.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k W_{jk} \quad (16)$$

Selanjutnya dikalikan dengan turunan dari fungsi aktivasinya untuk menghitung galat.

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (17)$$

kemudian menghitung perbaikan bobot (digunakan untuk memperbaiki  $v_{ij}$ )

$$\Delta v_{ij} = \delta_j x_i \quad (18)$$

kemudian menghitung perbaikan bias (untuk memperbaiki  $v_{0j}$ )

$$\Delta v_{0j} = \delta_j \quad (19)$$

Perbaikan bobot dan bias :

Masing-masing keluaran unit ( $y_k$ ,  $k = 1, \dots, m$ ) diperbaiki bias dan bobotnya

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \quad (20)$$

masing-masing unit tersembunyi ( $Z_j$ ,  $j = 1, \dots, p$ ) diperbaiki bias dan bobotnya

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \quad (21)$$

Selanjutnya proses yang terjadi adalah proses iterasi dalam dengan syarat penghentian yang ditentukan (Fausett, 1994).

Keterangan :

- $X_i$  : unit ke-i pada lapisan masukan
- $x_i$  : nilai aktivasi dari unit  $X_i$
- $Z_j$  : unit ke-j pada lapisan tersembunyi
- $Z_{in_j}$  : keluaran untuk unit  $Z_j$
- $z_j$  : nilai aktivasi dari unit  $Z_j$
- $Y_k$  : unit ke-k pada lapisan keluaran
- $Y_{in_k}$  : net masukan untuk unit  $Y_k$
- $y_k$  : nilai aktivasi dari unit  $Y_k$

- $w_{k0}$  : nilai bobot pada bias untuk unit  $Y_k$   
 $w_{kj}$  : nilai bobot dari  $Z_{ij}$  ke unit  $Y_k$   
 $\Delta w_{kj}$  : selisih antara  $w_{kj}(t)$  dengan  $w_{kj}(t+1)$   
 $v_{i0}$  : nilai bobot pada bias untuk unit  $Z_i$   
 $v_{ij}$  : nilai bobot dari unit  $X_i$  ke unit  $Z_i$   
 $\Delta v_{ij}$  : selisih antara  $v_{ij}(t)$  dengan  $v_{ij}(t+1)$   
 $\ddot{a}_k$  : faktor pengaturan nilai bobot pada lapisan keluaran  
 $\ddot{a}_j$  : faktor pengaturan nilai bobot pada lapisan tersembunyi  
 $\dot{a}$  : konstanta laju pelatihan (*learning rate*)  $0 < \dot{a} < 1$

### III. METODOLOGI PENELITIAN

#### 3.1. Objek dan Data Penelitian

Area yang dipilih untuk penelitian ini adalah daerah aliran sungai (DAS) Sungai Bengawan Solo, khususnya pada stasiun Jurug. Bagian hulu dimulai dari Bendungan Wonogiri hingga Kali Madiun, bagian setelah itu merupakan hilir sungai. Terdapat lima stasiun pengukur debit dan tinggi muka air di bagian hulu yang dioperasikan oleh Perum Jasa Tirta I, antara lain Juranggempal, Bendungan Colo, Serenan, Jurug, dan Tangen, serta satu stasiun pengukur curah hujan di Pabelan. Sungai Bengawan Solo memiliki enam anak sungai yang bermuara padanya mulai dari hulu hingga stasiun Jurug. DAS ini memiliki dua musim yaitu musun hujan (November – April) dan musun kemarau (Mei – Oktober). Data yang digunakan pada perancangan JST adalah data sekunder tinggi muka air rata-rata harian dalam satuan meter yang diukur pada stasiun Jurug pada tahun 2006 dan data curah hujan harian dalam satuan milimeter yang diperoleh dari stasiun Pabelan pada tahun 2006. Data yang ada dibagi menjadi dua kelompok, bulan Januari hingga November untuk pelatihan jaringan dan bulan Desember untuk pengujian. Karakteristik data tinggi muka air dan curah hujan dapat dilihat pada Tabel 1.

**Tabel 1. Statistik deskriptif data pelatihan dan pengujian JST**

Nilai	TMA (m)		Curah hujan (mm)		TMA & curah hujan	
	Data Pelatihan (n = 337)	Data Pengujian (n = 30)	Data Pelatihan (n = 337)	Data Pengujian (n = 30)	Data Pelatihan (n = 674)	Data Pengujian (n = 60)
Maksimum	6,33	6,3	92	85	92	85
Minimum	1,5	1,5	0	0	0	0
Rata-rata	2,53	2,45	5,51	12,92	4,02	7,69
Standard deviasi	6,33	0,85	14,04	19,42	10,06	14,68

### 3.2. Pengembangan Model

Dalam penelitian ini, dikembangkan tiga model JST yang kesemuanya menggunakan arsitektur *multilayer perceptron* dengan satu *hidden layer*. Arsitektur tersebut dipilih karena *hidden layer* diperlukan dalam mengidentifikasi fungsi *rainfall-runoff* yang sudah dikenal memiliki hubungan non-linear. Model tersebut diberi label BP1, BP2, dan BP3. *Input neuron* berisi data tinggi muka air rata-rata harian dan curah hujan harian beberapa hari sebelumnya. Ketiga model dilatih dengan jumlah *hidden neuron* yang bervariasi [3..10]. Output yang dihasilkan adalah prakiraan tinggi muka air harian.

Fungsi aktivasi yang digunakan pada studi ini adalah fungsi aktivasi hiperbolik tangen sigmoid pada *hidden layer*,

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

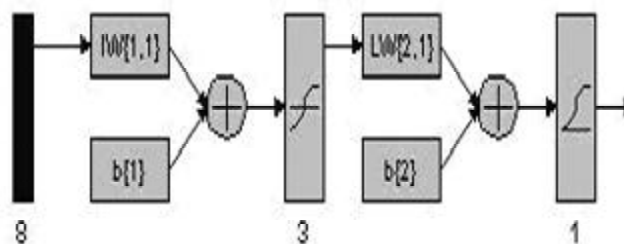
dan fungsi aktivasi logsigmoid pada *output layer*,

di mana  $x$  adalah elemen vektor dari lapisan sebelumnya. Kedua fungsi aktivasi dipilih karena mereka menghasilkan hasil yang baik dan menunjukkan fleksibilitas tinggi tanpa perubahan parameter model yang berarti (Alvisi, 2006). Karena fungsi logsigmoid memiliki nilai maksimum = 1, maka data input dan target  $p_i$  ditransformasikan sehingga semua polanya memiliki *range* yang sama dengan fungsi sigmoid yang dipakai melalui persamaan:

di mana  $[p_{min}, p_{max}]$  adalah variasi *range* dari variabel input.

Model BP1

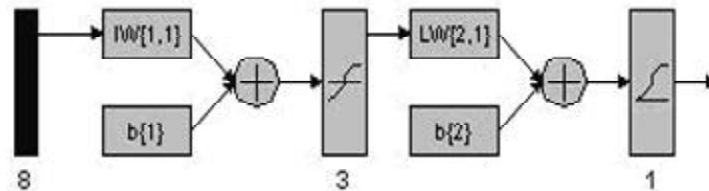
$$Am_{(t)} = f[Cm_{(t-4)}, Cm_{(t-3)}, Cm_{(t-2)}, Cm_{(t-1)}, Am_{(t-4)}, Am_{(t-3)}, Am_{(t-2)}, Am_{(t-1)}]$$



Gambar 7. Arsitektur BP1

## Model BP2

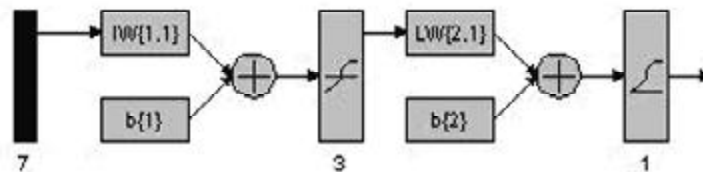
$$A_{(t)} = f[C_{(t-4)}, C_{(t-3)}, C_{(t-2)}, C_{(t-1)}, A_{(t-4)}, A_{(t-3)}, A_{(t-2)}, A_{(t-1)}]$$



Gambar 8. Arsitektur BP2

## Model BP3

$$dA_{(t)} = f[C_{(t-4)}, C_{(t-3)}, C_{(t-2)}, C_{(t-1)}, dA_{(t-3)}, dA_{(t-2)}, dA_{(t-1)}]$$



Gambar 9. Arsitektur BP3

Keterangan :

- Am: tinggi muka air (m)
- Cm : curah hujan (mm)
- A : tinggi muka air (dm)
- C : curah hujan (cm)
- t : waktu (hari)
- dA : selisih tinggi muka air (dm)

Model BP1 memiliki delapan input neuron terdiri atas empat data curah hujan (mm) dan empat data tinggi muka air (m) empat hari berturut-turut sebelum hari yang diperkirakan. Jumlah data *antecedent* sebanyak empat hari dipilih berdasarkan penelitian Elena Toth dan Armando Brath [2000] serta Rosmina Ahmad Bustami et al [2006]. Variabel input untuk model ini memiliki *range* [0..92].

Model BP2 hampir sama dengan BP1, hanya data curah hujan dalam satuan centimeter dan data tinggi muka air dalam satuan desimeter sehingga range antara nilai maksimum dan minimum data lebih sempit dari model BP1 [0..63,3].

Pada model BP3 digunakan selisih tinggi muka air dua hari berurutan dengan maksud *error* pada jaringan hanya berpengaruh pada selisih tinggi muka air yang bernilai lebih kecil dari pada nilai tinggi muka air secara keseluruhan dengan range input [-22,9..25]. Ketika variabel output berupa selisih tinggi muka air, prakiraan tinggi muka air adalah  $A_{(t-1)}$ ,  $dA_{(t)}$  dan *error* relatif hanya berpengaruh pada komponen kedua ( $dA_{(t)}$ ). Sebaliknya, ketika variabel output langsung berupa tinggi muka air  $A_{(t)}$ , *error* relatif yang sama akan menghasilkan *error* absolut yang lebih besar.

### 3.3 . Mekanisme Pelatihan

Algoritma pelatihan yang digunakan adalah Levenberg-Marquardt yang bekerja optimal pada model JST dengan sedikit *neuron* (Alvisi, 2006). *Learning rate* ( $\alpha$ ) yang digunakan dalam pelatihan 0,1 karena nilai *learning rate* yang terlalu besar dapat membuat algoritma menjadi tidak stabil dan mencapai titik minimum lokal (Siang, 2005). Nilai bobot dan bias awal diisi dengan bilangan acak kecil yang ditetapkan oleh MATLAB®. Bobot yang menghasilkan nilai turunan aktivasi yang kecil sedapat mungkin dihindari karena akan menyebabkan perubahan bobotnya menjadi sangat kecil. Demikian pula nilai bobot awal tidak boleh terlalu besar karena nilai turunan fungsi aktivasinya menjadi sangat kecil juga. Oleh karena itu, dalam standard *backpropagation*, bobot dan bias diisi dengan bilangan acak kecil (Siang, 2005).

Penilaian terhadap pelatihan jaringan dilakukan berdasarkan *mean squared error* (*mse*) dengan rumus :

$$MSE = \frac{1}{N} \sum_{i=1}^N (Q_i - \hat{Q}_i)^2 \quad (14)$$

Setiap jaringan dilatih hingga gradien perubahan *mse* mendekati nilai minimum, yang berarti nilai *mse* optimal untuk jaringan tersebut sudah tercapai. Untuk penghentian iterasi ditetapkan dengan jumlah *epoch* sebanyak 1000. Iterasi dilanjutkan lagi apabila nilai gradien perubahan *mse* belum mendekati nilai minimum. Proses pelatihan jaringan dilakukan menggunakan *software* MATLAB® 7.0.1.

### 3.4 . Mekanisme Pengujian

Semua model JST yang telah dilatih diuji dengan data uji, yaitu data tinggi muka air stasiun Jurug dan curah hujan stasiun Pabelan bulan Desember 2006. Nilai tinggi muka air prakiraan JST dan data tinggi muka air aktual dihitung korelasinya. Model jaringan yang menghasilkan nilai korelasi yang terbesar dan *error* absolut yang terkecil dipilih sebagai model jaringan yang paling optimal.

Nilai *error* yang kecil merepresentasikan ketepatan prediksi JST. Namun, nilai *error* sebaiknya tidak digunakan sebagai satu-satunya kriteria untuk menentukan performa jaringan (Yao, 1999). Oleh karena itu, digunakan juga nilai korelasi antara nilai tinggi muka air prakiraan JST dengan nilai tinggi muka air aktual sehingga didapatkan JST yang memiliki ketepatan prediksi dan korelasi yang tinggi dengan nilai aktual. Proses pengujian jaringan dilakukan dalam *software* MATLAB® 7.0.1, sementara korelasi data prakiraan dan data sebenarnya dilakukan dalam *software* MINITAB® 14.

#### IV. HASIL DAN PEMBAHASAN

##### 4.1 Hasil Pelatihan JST

###### Model BP1

Hasil pelatihan model BP1 dapat dilihat dalam Tabel 2.

Tabel 2. Hasil pelatihan model BP1

<i>Hidden neuron</i>	<i>mse</i>	<i>Gradien</i>
3	0,85633800	1,93315e-012
4	0,85633800	0
5	0,00001121	3,0342e-007
6	0,00001035	2,08031e-006
7	0,00000645	1,10774e-005
8	0,00000546	1,28962e-006
9	0,00000393	1,29396e-007
10	0,00000301	7,03059e-006

###### Model BP2

Hasil pelatihan model BP2 dapat dilihat dalam Tabel 3.

Tabel 3. Hasil pelatihan model BP2

<i>Hidden neuron</i>	<i>mse</i>	<i>Gradien</i>
3	0,00303727	0,000998285
4	0,00199912	9,52685e-010
5	0,00235055	1,88155e-006
6	0,00159447	1,33604e-006
7	0,00174382	7,25357e-005
8	0,00109690	0,000222114
9	0,18480800	2,5893e-032
10	0,00041502	2,08228e-005

**Model BP3**

Hasil pelatihan model BP3 dapat dilihat dalam Tabel 4.

Tabel 4. Hasil pelatihan model BP3

<i>Hidden neuron</i>	<i>mse</i>	<i>Gradien</i>
3	4,96504e-005	1,08868e-00
4	5,69474e-005	2,43572e-009
5	4,04911e-005	7,82241e-011
6	3,49326e-005	7,05494e-010
7	2,7522e-005	2,88485e-007
8	1,73947e-005	1,02691e-006
9	2,00082e-005	7,53881e-008
10	1,96478e-005	9,53003e-007

Perbandingan performa ketiga model JST dapat dilihat pada Tabel 5.

Tabel 5. Perbandingan performa pelatihan BP1, BP2, dan BP3

<i>Hidden neuron</i>	<b>Rata-rata <i>error</i> absolut</b>		
	<b>BP1 (dm)</b>	<b>BP2 (dm)</b>	<b>BP3 (dm)</b>
3	945.95	3.8762	0.3750
4	945.95	3.1447	0.4016
5	3.4225	3.4099	0.3387
6	3.2886	2.8085	0.3146
7	2.5961	2.9371	0.2792
8	2.3886	2.3294	<b>0.2220</b>
9	2.0265	30.236	0.2381
10	<b>1.7735</b>	<b>1.4328</b>	0.2359

Dari tabel di atas dapat dilihat bahwa dengan jumlah *hidden neuron* yang semakin banyak, *error* absolut jaringan pada fase pelatihan cenderung semakin kecil.

**4.2. Pengujian JST**

Pengujian ketiga model JST dilakukan dengan 27 set data tinggi muka air stasiun Jurug dan curah hujan stasiun Pabelan bulan Desember 2006. Pengujian dilakukan dalam lingkungan *software* MATLAB® 7.0.1. Parameter yang dinilai dalam proses pengujian

antara lain, Korelasi, dan *mse* nilai tinggi muka air prakiraan JST dengan data sebenarnya.

### Model BP1

Hasil pengujian model BP1 dapat dilihat dalam Tabel 6.

Tabel 6. Hasil pengujian model BP1

<i>Hidden neuron</i>	<i>mse</i>	Korelasi
3	<b>0,0080501</b>	-
4	<b>0,0080501</b>	-
5	9,2992000	<b>0,395</b>
6	7,47400001	0,434
7	1,318700030	0,277
8	6,8811000	-0,100
9	9,8872000	0,275
10	7,3221000	0,287

### Model BP2

Hasil pengujian model BP2 dapat dilihat dalam Tabel 7.

Tabel 7. Hasil pengujian model BP2

<i>Hidden neuron</i>	<i>mse</i>	Korelasi
3	778,7632	<b>0,754</b>
4	769,4621	0,569
5	619,8831	0,516
6	749,6946	0,577
7	909,6758	0,589
8	715,7946	0,409
9	<b>11,5978</b>	-
10	780,6696	0,350



**Model BP3**

Hasil pengujian model BP3 dapat dilihat dalam Tabel 8.

Tabel 8. Hasil pengujian model BP3

<i>Hidden neuron</i>	<i>mse</i>	<i>Korelasi</i>
3	103,5014	-0,280
4	<b>37,6029</b>	0,011
5	117,2173	0,385
6	162,1188	<b>0,428</b>
7	47,4640	0,089
8	54,4122	0,073
9	100,7758	0,424
10	393,0124	0,121

Dari ketiga tabel hasil pengujian di atas dapat dilihat bahwa nilai *mse* model BP2 dan BP3 lebih besar dari model BP1. Hal itu dikarenakan output dari model BP2 dan BP3 berupa prakiraan tinggi muka air dalam satuan desimeter, sedangkan output model BP1 dalam satuan meter sehingga nilai *error* relatif dari model BP2 dan BP3 lebih besar.

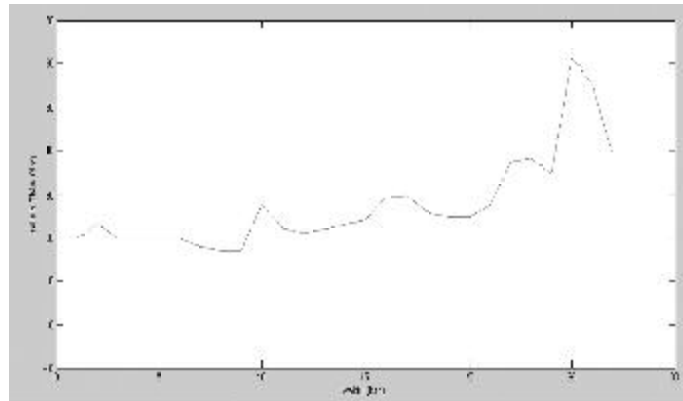
Perbandingan performa jaringan pada fase pengujian dapat dilihat pada Tabel 9.

Tabel 9. Perbandingan peforma pengujian BP1, BP2, dan BP3

<i>Hidden neuron</i>	<b>BP1 (dm)</b>		<b>BP2 (dm)</b>		<b>BP3 (dm)</b>	
	Rata-rata <i>error</i> absolut (dm)	<i>korelasi</i>	Rata-rata <i>error</i> absolut (dm)	<i>korelasi</i>	Rata-rata <i>error</i> absolut (dm)	<i>korelasi</i>
3	897.222	-	27.906	<b>0,754</b>	10.174	-0,280
4	897.222	-	27.739	0,569	<b>6.1321</b>	0,011
5	30.495	<b>0,395</b>	24.897	0,516	10.8270	0,385
6	27.339	0,434	27.381	0,577	12.7330	<b>0,428</b>
7	33.643	0,277	30.161	0,589	6.8894	0,089
8	175.180	-0,100	26.754	0,409	7.3765	0,073
9	31.444	0,275	<b>3.4056</b>	-	10.0390	0,424
10	<b>27.059</b>	0,287	27.94	0,350	19.8250	0,121

Pada hasil pengujian model BP2 dengan 9 *hidden neuron* (BP2-9) didapatkan nilai

*mse* terkecil dibandingkan model BP2 dengan jumlah *hidden neuron* berbeda, namun nilai korelasinya tidak bisa didapatkan karena hasil prakiraan BP2-9 untuk berbagai input yang berbeda menghasilkan nilai prakiraan sama. Grafik perbandingan nilai prakiraan BP2-9 dengan nilai aktual dapat dilihat pada Gambar 10.

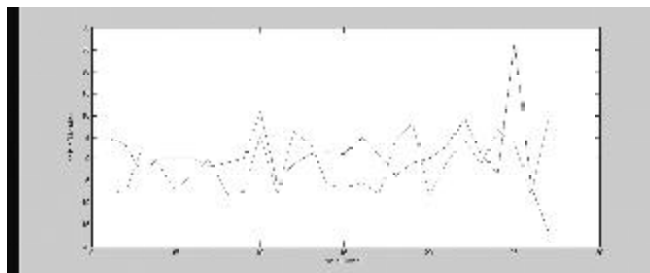


Gambar 10. Perbandingan nilai prakiraan BP2-9 dengan nilai aktual

#### 4.3 . Pemilihan Model JST yang Optimal

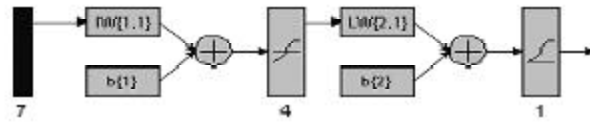
Dari Tabel 5. dan Tabel 9. dapat dilihat bahwa nilai *error* yang kecil pada tahap pelatihan belum tentu akan menghasilkan nilai *error* yang kecil juga pada tahap pengujian. Hal itu karena jaringan yang lebih kompleks (dengan banyak *hidden neuron*) dan terlalu terfiksir (*overtrained*) pada data pelatihan dapat menghasilkan *error* yang kecil pada tahap pelatihan, namun jaringan tersebut dapat menemui kesulitan dalam beradaptasi dengan data lain (Yao, 2001).

Berdasarkan nilai *error* absolut yang terkecil, maka model JST yang dipilih untuk prakiraan tinggi muka air stasiun Jurug adalah model BP3 dengan 4 *hidden neuron* (BP3-4). Model tersebut menghasilkan nilai *mse* 37,5014 atau *error* absolut sebesar 6,13 dm dan korelasi sebesar 0,011. Grafik perbandingan nilai prakiraan BP3-4 dengan hasil pengukuran di lapangan dapat dilihat pada Gambar11.



Gambar 11. Perbandingan nilai prakiraan BP3-4 dengan nilai aktual

Berikut parameter-parameter model BP3 dengan 4 *hidden neuron* :



Gambar 12. Arsitektur BP3-4

Input	: $C_{(t-4)}, C_{(t-3)}, C_{(t-2)}, C_{(t-1)}, dA_{(t-3)}, dA_{(t-2)}, dA_{(t-1)}$
Output	: $dA_{(t)}$
Hidden layer	: 1
Hidden neuron	: 4
Fungsi aktivasi	
Hidden layer	: tansig
Output layer	: logsig
Bias input layer	: [-16.3337; -1.0365; 165.3072; 349.2208]
Bias hidden layer	: [-33.2394]
Bobot input layer	: [5.5298 5.9742 2.7682 7.2158 5.9564 -10.2871 11.6604; 1.9651 4.4661 -2.4715 6.8661 1.1737 - 5.90686.7718; 29.2629 38.3369 53.3028 -74.3715 -38.2071 - 51.8178 -225.5871; 1.5837 24.1472 3.0253 - 73.9591 -63.0203 -27.9642 -592.5223]
Bobot hidden layer	: [-0.090101 70.7996 -37.7457]

## I. KESIMPULAN

- Dari hasil pelatihan dan pengujian ketiga model JST dengan parameter jaringan yang berbeda-beda, didapatkan model yang paling optimal untuk prakiraan tinggi muka air Sungai Bengawan Solo Stasiun Jurug, yaitu *multiplayer perceptron* dengan satu *hidden layer* dan empat *hidden neuron* (BP3-4) karena model tersebut menghasilkan nilai *error* terkecil dibandingkan model yang lain.
- Penggunaan selisih tinggi muka air sebagai input dan output pada BP3 menghasilkan performa jaringan yang lebih baik dibandingkan dengan penggunaan tinggi muka air sebagai input dan output.
- Perbedaan jumlah *hidden neuron* berpengaruh pada kemampuan jaringan dalam mengenali pola dan beradaptasi dengan pola baru. Semakin banyak *hidden neuron*, semakin tinggi kemampuan jaringan dalam mengenali pola data pelatihan. Namun jumlah *hidden neuron* yang terlalu besar menurunkan kemampuan adaptasi jaringan

terhadap pola baru. Oleh karena itu diperlukan pelatihan dan pengujian dengan jumlah *hidden neuron* yang berbeda-beda untuk menentukan jumlah *hidden neuron* yang sesuai.

d. Dari hasil pengujian ketiga model JST, diperoleh arsitektur optimal JST untuk prakiraan tinggi muka air Sungai Bengawan Solo yaitu *multilayer perceptron* dengan satu *hidden layer* dan empat *hidden neuron* (BP3-4) dengan parameter-parameter jaringan sebagai berikut :

Input :  $C_{(t-4)}, C_{(t-3)}, C_{(t-2)}, C_{(t-1)}, dA_{(t-3)}, dA_{(t-2)}, dA_{(t-1)}$   
 Output :  $dA_{(t)}$   
 Hidden layer : 1  
 Hidden neuron : 4  
 Fungsi aktivasi  
 Hidden layer : tansig  
 Output layer : logsig  
 Bias input layer : [-16.3337; -1.0365; 165.3072; 349.2208]  
 Bias hidden layer : [-33.2394]  
 Bobot input layer : [5.5298 5.9742 2.7682 7.2158 5.9564 -10.2871  
 11.6604;  
 1.9651 4.4661 -2.4715 6.8661 1.1737 -5. 9068  
 6.7718;  
 29.2629 38.3369 53.3028 -74.3715 -38.2071 –  
 51.8178 -225.5871;  
 1.5837 24.1472 3.0253 -73.9591 -63.0203 -27.9642  
 -592.5223]  
 Bobot hidden layer : [-0.090101 70.7996 -37.7457 -0.035552 ]

## II. DAFTAR PUSTAKA

1. Alvisi, S., et al, (2006). *Water Level Forecasting through Fuzzy Logic and Artificial Neural Network Approaches*. Hydrology and Earth System Sciences, 10, hal : 1-17.
2. Aqil, Muhammad, et al, (2006). *Decision Support System for Flood Crisis Management using Artificial Neural Network*. International Journal of Intelligent Technology, Volume 1, no. 1, hal : 70-6.
3. Bustami, Rosmina Ahmad, et al, (2006). *Artificial Neural Network for Daily Water Level Forecasting Estimation*. Engineering e-Transaction, Vol. 1, No. 1, page : 7-12.

4. Danh, Nguyen Tan, et al, (1999). *Neural Network Models for River Flow Forecasting*. Water SA, Vol. 25, No. 1, page : 33-9.
5. Dibike, Yonas B., Dimitri P. Solomatine, (2000). *River Flow Forecasting Using Artificial Neural Network*. EGS Journal of Physics and Chemistry of the Earth.
6. Fausett, L., (1994). *Fundamentals of Neural Networks, Architecture, Algorithms, and Applications*. New Jersey, Prentice Hall Inc.
7. Johannet, A., et al, (2005). *Neural Network : From Black Box towards Transparent Box Application to Evapotranspiration Modeling*. International Journal of Computational Intelligence, Vol. 4, No. 3, hal : 163-70.
8. Siang, Jong Jek, (2005). *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Yogyakarta, Penerbit Andi.
9. Toth, Elena, Armando Brath, (2000). *Flood Forecasting Using Artificial Neural Network in Black-Box and Conceptual Rainfall-Runoff Modelling*. Diakses pada 10 januari (2008) dari : [www.cig.ensmp.fr/~iahs/hsj/470/hysj\\_47\\_06\\_0865.pdf](http://www.cig.ensmp.fr/~iahs/hsj/470/hysj_47_06_0865.pdf)
10. Yani, Eli, (2005). *Pengantar Jaringan Syaraf Tiruan*. Diakses pada 10 januari 2008 dari [www.MateriKuliah.com](http://www.MateriKuliah.com).
11. Yao, Jingtao, et al, (1999). *Neural Network for Technical Analysis: a Study on KLCI*. International Journal of Teoretical and Applied Finance, Vol. 2, No. 2, page : 221-41.42
12. Yao, Jingtao, Chew Lim Tan, 2001. *Guidelines for Financial Forecasting with Neural Network*. Conference on Neural Information Processing, Shanghai, China, hal : 757-61.