

# Sistem Informasi Penyebaran Penyakit Demam Berdarah Menggunakan Metode Jaringan Syaraf Tiruan *Backpropagation*

Didi Supriyadi

Program Studi Teknik Informatika

Sekolah Tinggi Teknologi Telematika Telkom Purwokerto

Banyumas – Jawa Tengah

didi\_supriyadi121105@yahoo.co.id

**Abstrak**—Penyakit Demam Berdarah salah satu masalah kesehatan utama dan endemik di beberapa Negara. Indonesia termasuk dalam kategori “A” dalam stratifikasi DBD oleh WHO tahun 2001 yang mengindikasikan tingginya angka perawatan pada rumah sakit dan kematian akibat DBD. Tujuan penelitian ini untuk menginvestigasi kemampuan metode Jaringan syaraf tiruan *Backpropagation* untuk prediksi penyebaran penyakit demam berdarah pada suatu wilayah. Penelitian ini menggunakan enam variabel input yang merupakan faktor lingkungan yang mempengaruhi penyebaran penyakit demam berdarah, yaitu suhu rata – rata, curah hujan, jumlah hari hujan, kepadatan penduduk, ketinggian permukaan air laut, dan persentase angka bebas jentik. Arsitektur jaringan yang diterapkan jaringan dengan *multilayer* yang menggunakan sebuah input dengan 6 neuron, satu *hidden layer* dan sebuah output dengan jumlah neuron output adalah satu. Dari hasil pelatihan diperoleh arsitektur jaringan terbaik adalah dengan jumlah satu *hidden layer* dengan jumlah neuron sebanyak 110 neuron dan diperoleh sistem dapat mengenali seluruh data pelatihan. Algoritma pelatihan terbaik dengan menggunakan variabel pesat belajar sebesar 0,9 dan momentum sebesar 0,6 dengan MSE 0.000999879, pada proses pengujian menggunakan 17 data pengujian diperoleh tingkat akurasi jaringan sekitar 88,23% dan tingkat kesalahannya sekitar 11,77%.

**Kata kunci**—Jaringan Syaraf Tiruan; *Backpropagation*; Demam Berdarah

## I. PENDAHULUAN

Demam berdarah *dengue* (DBD) salah satu permasalahan utama yang berkembang dan endemik di benua Amerika, Eropa, Afrika, Asia, Australia, beberapa pulau di India, dan Caribbia [4]. *World Health Organization* (WHO) mencatat sekitar 50 – 100 juta kasus *dengue* yang direkam dari semua lapisan dunia, dan 2/5 populasi dunia memiliki resiko tinggi terhadap *dengue* dan lebih dari seratus negara telah terinfeksi virus *dengue* [4]. Indonesia termasuk dalam kategori “A” dalam stratifikasi DBD oleh WHO tahun 2001 yang mengindikasikan tingginya angka perawatan pada rumah sakit dan kematian akibat DBD, khususnya pada anak [3].

Tujuan dari penelitian ini adalah untuk menginvestigasi kemungkinan penerapan Jaringan Syaraf Tiruan (JST) sebagai teknik untuk prediksi penyebaran penyakit demam berdarah

secara dini berdasarkan parameter : suhu rata – rata, kepadatan penduduk (populasi), curah hujan, jumlah hari hujan, %ABJ dan ketinggian dari permukaan air laut dan laporan kasus *dengue* sebagai koresponden keluaran dari parameter tersebut.

## II. KERANGKA TEORI

### A. Demam Berdarah Dengue (DBD)

DBD salah satu penyakit menular berbahaya yang dapat menimbulkan kematian dalam waktu singkat dan wabah yang disebabkan virus *dengue*. Penyakit ini ditularkan melalui gigitan nyamuk *Aedes aegypti* dan mungkin juga *Albopictus*. Kedua jenis nyamuk ini terdapat hampir di seluruh Indonesia kecuali ketinggian lebih dari 1000 meter diatas permukaan laut [11]. Simptom *dengue* terjadi secara mendadak dan ditandai dengan demam tinggi, sakit kepala, pusing, tulang dan sendi terasa nyeri, lidah terasa tidak enak, dan wajah pucat [2]. Pada kulit penderita terdapat bercak merah yang diakibatkan oleh gigitan nyamuk [1].

Tahun 2002 jumlah kasus sebanyak 40.377 (*incidence rate* (IR) : 19,24/100.000 penduduk dengan 533 kematian (*case fatality rate* (CFR) : 1,3%), tahun 2003 jumlah kasus sebanyak 52.566 (IR : 24,34/100.000) dengan 814 kematian (IR : 1,20%). Sampai dengan tahun 2007, jumlah kasus maupun jumlah kematian terus meningkat yaitu mencapai 124.811 (IR : 57,52/100.000 penduduk) dengan angka kematian 1.277 (CFR : 1,02%)[5].

### B. Jaringan Syaraf Tiruan (JST)

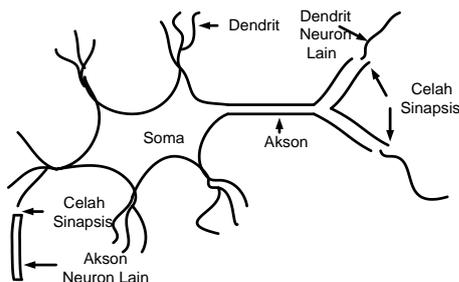
JST adalah model komputasi dari otak manusia [1]. JST merupakan suatu sistem pemrosesan informasi yang mempunyai karakteristik baik struktur dan cara kerja menyerupai jaringan syaraf manusia. Hal ini dikarenakan manusia memiliki banyak keunggulan dibandingkan makhluk lain, otak manusia memiliki struktur yang sangat kompleks dan mampu berpikir yang dapat memecahkan persoalan yang dihadapinya dan mampu belajar dari pengalaman masa lalu. Jaringan syaraf biologis pada otak manusia terdiri dari sel-sel syaraf yang disebut *neuron* yang saling berhubungan satu dengan yang lain, pada suatu penghubung yang disebut sinapsis.

Tiga komponen penting pada sel syaraf biologis (Gambar 1) yang digunakan untuk memahami JST, [9]: (1) Dendrit, merupakan elemen pemrosesan yang menerima dan melewatkan sinyal masukan dari neuron lain. Sebuah neuron mampu menerima 5.000 sampai 15.000 sinyal masukan. Sinyal tersebut dimodifikasi dengan bobot (diperkuat/diperlemah) pada sinapsis penerima, (2) Soma/badan sel, berfungsi mengakumulasi sinyal masukan terbobot yang dilewatkan melalui dendrit. Jika sinyal – sinyal tersebut lebih besar dari batas ambang tertentu (*threshold*), maka sel akan dipicu sehingga akan mentransmisikan ke neuron lain, (3) Akson, berfungsi sebagai saluran keluaran dari suatu neuron yang akan menyalurkan sinyal ke neuron yang lain.

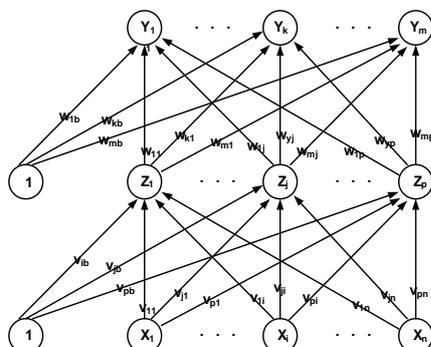
**C. JST Backpropagation**

Pelatihan *JST Backpropagation* biasanya dilakukan dengan cara diawasi [7]. Dalam proses pembelajaran, arsitektur JST pemetaan input – output, menyesuaikan bobot dan bias pada setiap iterasi didasarkan pada minimalisasi atau optimasi dari beberapa kesalahan yang diukur antara output yang dihasilkan dan output yang diinginkan. Proses ini diulang sampai kriteria konvergensi diterima dan tercapai [1].

Arsitektur *JST Backpropagation* memiliki beberapa unit yang ada dalam satu atau lebih lapis tersembunyi. Gambar 2 adalah arsitektur *backpropagation* dengan *n* buah masukan dimana  $1 < i < n$  ditambah sebuah bias, sebuah *hidden layer* yang terdiri dari *p* unit dimana  $1 < j < p$  ditambah sebuah bias, serta *m* buah unit keluaran dimana  $1 < k < m$ .  $v_{ji}$  merupakan bobot garis dari unit masukan  $x_i$  ke unit lapis tersembunyi  $z_j$  ( $v_{jb}$  merupakan bobot garis yang menghubungkan bias di unit tersembunyi  $z_j$ ).  $w_{kj}$  merupakan bobot dari unit lapis tersembunyi  $z_j$  ke unit keluaran  $y_k$  ( $w_{kb}$  merupakan bobot dari bias di lapis tersembunyi ke unit keluaran  $y_k$ ).



Gambar 1. Neuron Tunggal



Gambar 2. Arsitektur JST Backpropagation

Algoritma pelatihan untuk jaringan dengan satu lapis tersembunyi (dengan fungsi aktivasi *sigmoid* biner) adalah sebagai berikut [7]:

- L0 : Inialisasi bobot dengan bilangan acak kecil.
- L1 : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 – 8.
- L2 : Untuk setiap pasang data pelatihan, lakukan langkah 3-8.
- FI : Feed Forward**
- L3 : Tiap unit input menerima sinyal dan meneruskannya ke unit tersembunyi diatasnya.
- L4 : Hitung semua output di unit tersembunyi  $z_j$  ( $j=1, 2, \dots, p$ )

$$z\_net_j = v_{j0} + \sum_{i=1}^n x_i v_{ji} \tag{1}$$

$$z_j = f(z\_net_j) = \frac{1}{1 + e^{-z\_net_j}} \tag{2}$$

- L5 : Hitung semua output jaringan di unit  $y_k$  ( $k = 1, 2, \dots, m$ )

$$y\_net_k = w_{k0} + \sum_{j=1}^p z_j w_{kj} \tag{3}$$

$$y_k = f(y\_net_k) = \frac{1}{1 + e^{-y\_net_k}} \tag{4}$$

**FII : Backpropagation**

- L6 : Hitung faktor  $\delta$  unit keluaran berdasarkan kesalahan di setiap unit keluaran  $y_k$  ( $k = 1, 2, \dots, m$ )

$$\delta_k = (t_k - y_k) f'(y\_net_k) = (t_k - y_k) y_k (1 - y_k) \tag{5}$$

$\delta_k$  merupakan unit kesalahan yang akan dipakai dalam perubahan bobot lapis dibawahnya (langkah 7). Hitung suku perubahan bobot  $w_{kj}$  (yang akan dipakai untuk merubah bobot  $w_{kj}$ ) dengan laju percepatan  $\alpha$

$$\Delta w_{kj} = \alpha \cdot \delta_k \cdot z_j \quad (k = 1, 2, \dots, m; j = 0, 1, \dots, p) \tag{6}$$

- L7 : Hitung faktor  $\delta$  unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi  $z_j$  ( $j = 1, 2, \dots, p$ )

$$\delta\_net_j = \sum_{k=1}^m \delta_k w_{kj} \tag{7}$$

faktor  $\delta$  unit tersembunyi:

$$\delta_j = \delta\_net_j \cdot f'(z\_net_j) = \delta\_net_j \cdot z_j (1 - z_j) \tag{8}$$

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

$$\Delta v_{ji} = \alpha \cdot \delta_j \cdot x_i \quad (j = 1, 2, \dots, p; i = 0, 1, \dots, n) \tag{9}$$

**FIII : Perubahan Bobot**

- L8 : Hitung semua perubahan bobot Perubahan bobot garis yang menuju ke unit keluaran:

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (k = 1, 2, \dots, m; j = 0, 1, \dots, p) \tag{10}$$

Perubahan bobot garis menuju ke unit tersembunyi:

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (j=1, 2, \dots, p; i = 0, 1, \dots, n) \tag{11}$$

- L9 : Selesai

Berdasarkan algoritma di atas dapat diketahui bahwa pelatihan *backpropagation* meliputi 3 fase yang akan diulang-ulang terus hingga kondisi penghentian dipenuhi [9].

a. Fase Feed Forward/Umpan Maju

Selama propagasi maju, sinyal masukan ( $= x_i$ ) dipropagasikan ke lapis tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit lapis

tersembunyi (=  $z_j$ ) tersebut selanjutnya dipropagasikan maju lagi ke lapis tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan (=  $y_k$ ). Keluaran jaringan (=  $y_k$ ) dibandingkan dengan target yang harus dicapai (=  $t_k$ ). Selisih  $t_k - y_k$  adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Tetapi jika kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasikan untuk mengurangi kesalahan.

**b. Propagasi Balik**

Berdasarkan kesalahan  $t_k - y_k$ , dihitung faktor  $\delta_k (k=1, 2, \dots, m)$  yang dipakai untuk mendistribusikan kesalahan di unit  $y_k$  ke semua unit tersembunyi yang terhubung langsung dengan  $y_k$ .  $\delta_k$  juga dipakai untuk mengubah bobot garis yang menghubungkan langsung dengan unit keluaran.

Dengan cara yang sama, dihitung  $\delta_j$  di setiap unit di lapis tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di lapis di bawahnya. Demikian seterusnya hingga faktor  $\delta$  di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

**c. Perubahan Bobot**

Setelah semua faktor  $\delta$  dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor  $\delta$  neuron di lapis atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke lapis keluaran didasarkan atas dasar  $\delta_k$  yang ada di unit keluaran.

**D. Sum Square Error dan Root Mean Square Error**

Kesalahan pada keluaran jaringan merupakan selisih antara keluaran sebenarnya dengan keluaran yang diinginkan. Selisih yang dihasilkan antara keduanya biasanya ditentukan dengan cara dihitung menggunakan suatu persamaan [7].

Sum Square Error (SSE) dihitung sebagai berikut:

- a. Hitung keluaran jaringan syaraf untuk masukan pertama.
- b. Hitung selisih antara nilai keluaran jaringan syaraf dan nilai target / yang diinginkan untuk setiap keluaran.
- c. Kuadratkan setiap keluaran kemudian hitung seluruhnya.

Adapun rumusnya adalah:

$$SSE = \sum_p \sum_j (T_{jp} - X_{jp})^2 \tag{12}$$

dengan:

$T_{jp}$  = nilai keluaran jaringan syaraf

$X_{jp}$  = nilai target/yang diinginkan untuk setiap keluaran

Root Mean Square Error (RMS Error):

- a. Hitung SSE
- b. Hasilnya dibagi dengan perkalian antara banyaknya data pada pelatihan dan banyaknya keluaran, kemudian diakarkan.

$$RMS\ Error = \sqrt{\frac{\sum_p \sum_j (T_{jp} - X_{jp})^2}{n_p n_o}} \tag{13}$$

dengan:

$T_{jp}$  = nilai keluaran jaringan syaraf

$X_{jp}$  = nilai target/yang diinginkan untuk setiap keluaran

$n_p$  = jumlah seluruh pola

$n_o$  = jumlah keluaran

Keberhasilan proses belajar JST ditunjukkan dengan besarnya error yang minimum. Pada kondisi inilah JST tersebut dapat digunakan. Ketika ada hal baru yang harus diketahui oleh JST, maka proses belajar harus diulang kembali dengan menggunakan informasi yang lama ditambah dengan informasi yang baru.

**III. METODOLOGI**

**A. Metode Pengumpulan data**

Metode pengumpulan data meliputi metode studi literatur dan wawancara. Langkah – langkah pengumpulan data yang dilakukan pada penelitian ini meliputi:

- a. Membuat rancangan data input dan data output sebagai data pelatihan maupun data pengujian

Berdasarkan keterangan yang diperoleh dari ahli epidemiologi kabupaten Brebes (Awaludin, SKM., M.Kes), data yang digunakan dalam penelitian ini meliputi: suhu rata – rata ( $^{\circ}C$ ) dari BMKG Kabupaten Brebes yang menginduk di BMKG kota Tegal, data curah hujan (mm), data jumlah hari hujan (hh), data Kepadatan penduduk ( $km^2$ ), data ketinggian daratan dari permukaan air laut (m) diperoleh dari BPS Kabupaten Brebes, serta data persentase ABJ (%) dari Dinas Kesehatan Kabupaten Brebes. Dalam penelitian ini juga mengambil data kasus DBD yang digunakan sebagai target dari sistem yang diperoleh dari Dinas Kesehatan Kabupaten Brebes.

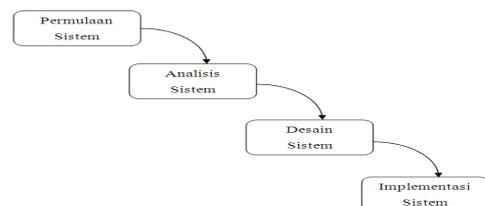
Data yang diambil secara *time series*, sehingga ada 102 pasangan data yang telah diperoleh merupakan data nilai tahunan dari 6 parameter tersebut yang meliputi 17 kecamatan di Kabupaten Brebes tahun 2005 – 2010.

- b. Membagi data ke dalam data set pelatihan dan data set pengujian

Dari 102 data dibagi menjadi dua kelompok, yaitu data yang berfungsi sebagai masukan dan keluaran untuk proses pelatihan dan pengujian. Sebanyak 85 set data variabel faktor lingkungan yang mempengaruhi penyebaran penyakit DBD dijadikan sebagai matriks dengan ukuran 7 x 85 untuk data pelatihan (data tahun 2005 - 2009) dan 17 set data digunakan sebagai data pengujian (data tahun 2010) dengan ukuran matriks 7 x 17.

**B. Metode pengembangan sistem**

Teknik pengembangan sistem yang dikembangkan oleh Whitten (Gambar 4) yang dijabarkan sebagai berikut [10]:



Gambar 4 Metode Sekuensial Waterfall  
Sumber : (Whitten, 2004)

a. Rekayasa dan Pemodelan Sistem

Untuk menggambarkan sistem informasi penyebaran penyakit DBD dengan metode JST Backpropagation yang akan dikembangkan, maka perlu dibuat diagram alir sistem.

b. Analisis

Untuk dapat memprediksi penyebaran penyakit DBD pada suatu wilayah saat ini masih terkendala pakar yang sangat terbatas. Sehingga masalah ini tidak jarang menyebabkan kejadian luar biasa (KLB). Adapun pertimbangan dalam penelitian ini menggunakan metode JST Backpropagation antara lain dikarenakan aturan pengetahuan untuk memprediksi penyebaran penyakit DBD yang sulit untuk dirumuskan. Akan tetapi dengan data historis/data kasus yang jumlahnya cukup besar maka JST layak untuk diuji coba untuk mengetahui kemampuan JST dalam menyelesaikan permasalahan penyebaran penyakit DBD.

c. Desain

Desain sistem dengan JST meliputi beberapa tingkatan yaitu: tingkat neuron (node), tingkat jaringan (banyaknya lapisan), dan tingkat pelatihan.

Rancangan jumlah neuron pada JST Backpropagation untuk sistem prediksi penyebaran penyakit DBD: (1) Jumlah neuron lapisan input sebanyak 6 neuron, (2) Jumlah lapisan hidden layer dan neuron penyusunnya ditentukan secara konstruktif, (3) Jumlah lapisan output terdiri atas satu neuron.

Penentuan 6 neuron pada lapisan input didasarkan pada data faktor lingkungan yang mempengaruhi penyebaran penyakit DBD yang diperoleh di lokasi penelitian.

Jumlah neuron untuk lapisan output adalah satu buah neuron dengan pertimbangan bahwa pada aplikasi prediksi penyebaran penyakit DBD digunakan untuk menentukan nilai kondisi suatu wilayah terkait ada dan tidaknya kasus DBD. Pada pengolahan data output, kondisi suatu wilayah terhadap kasus DBD direpresentasikan dengan bilangan biner 0 dan 1.

Tahap desain dimulai dari perancangan arsitektur jaringan seperti pada Gambar 5.

d. Implementasi

Nilai data input adalah bebas artinya nilai dapat diisi dengan sembarangan bilangan berdasarkan data yang diperoleh di lokasi penelitian. Data tersebut kemudian disusun dalam bentuk matriks dengan ukuran matriks 7 x 85 untuk data yang akan digunakan sebagai data pelatihan.

Data tersebut diolah dalam suatu range tertentu yang disebut normalisasi data. Runtun data masukan dan target dinormalisasi dengan membawa data ke bentuk normal yang memiliki mean = 0 dan deviasi standard = 1. Pada simulasi ini metode normalisasi yang digunakan berdasarkan rumus [8]:

$$A_i' = \frac{A_i - A_{\min}}{A_{\max} - A_{\min}} \quad (14)$$

$A_i$  merupakan bilangan ke-i dan  $A_{\max}$  adalah bilangan maksimum dalam suatu N (data) bilangan masukan serta  $A_i'$  adalah bilangan baru ke-i setelah proses normalisasi.

Setelah dinormalisasi maka proses selanjutnya adalah proses komputasi backpropagation dengan menerapkan semua rumus dan algoritma dari backpropagation.

Untuk mencapai tujuan ini perlu dipahami proses – proses yang terjadi di dalam JST Backpropagation. Proses tersebut meliputi: (1) Inisialisasi Bobot dan bias, (2) Penentuan kondisi berhenti, (3) Set pelatihan sebanyak 85 data (pasangan vektor input dan vektor target) akan menjalani tahap – tahap yang telah dijelaskan.

Langkah nomor 3 akan diulangi sampai kondisi berhenti terpenuhi sehingga akan memerlukan banyak iterasi sebelum proses pelatihan berhenti. Setelah membangun JST Backpropagation dan menentukan parameter jaringan langkah komputasi selanjutnya adalah pelatihan JST. Pelatihan dilakukan dengan variasi parameter jaringan sebagai berikut:

- Pelatihan 1, Mencari Jumlah Neuron untuk Hidden Layer

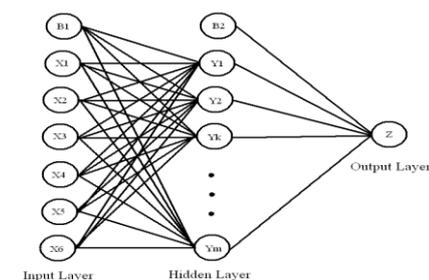
Tujuan dari langkah ini adalah menentukan jumlah neuron untuk hidden layer yang menghasilkan kinerja terbaik selama proses pelatihan, ditunjukkan dengan MSE terkecil dan waktu komputasi yang cepat. Pengamatan dilakukan terhadap jaringan dengan satu hidden layer, dengan variasi jumlah neuron hidden layer adalah 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 105 dan 110 neuron.

- Pelatihan 2, Mencari Momentum dan Laju Pembelajaran Terbaik

Pada langkah ini diamati variasi momentum dan pesat belajar terhadap unjuk kerja jaringan. Tujuan dari langkah ini adalah untuk memperoleh nilai momentum dan pesat belajar yang optimum. Dengan memakai arsitektur jaringan terbaik pada pelatihan 1. Kinerja tujuan (target error) yang diberikan adalah 0,001, maksimum iterasinya 15000 epoch dan algoritma pelatihannya gradient descent dengan momentum dan pesat belajar (traingdx) .

- Pengujian

Pengujian dapat dilakukan dengan menggunakan data set yang sudah dilatihkan untuk melihat unjuk kerja sistem aplikasi yang telah dibuat dengan melihat nilai error minimumnya. Selain itu juga pengujian dapat dilakukan menggunakan data set yang belum pernah dilatihkan sebelumnya untuk melihat tingkat akurasi sistem yang telah dibuat, yaitu menggunakan data uji sebanyak 17 pasangan data.



Gambar 5 Arsitektur JST untuk Sistem Informasi Penyebaran Penyakit DBD

#### IV. HASIL DAN PEMBAHASAN

##### A. Pelatihan

TABLE I. HASIL PELATIHAN DENGAN SATU HIDDEN LAYER UNTUK JUMLAH NEURON HIDDEN LAYER YANG DIVARIASI

Jumlah Neuron (nL1)	MSE	Waktu
5	0.0216945	00:49.70
10	0.0161268	00:52.83
15	0.028515	00:55.62
20	0.00705141	00:57.63
25	0.00352099	00:59.85
30	0.0253289	01:01.45
35	0.0244566	01:03.31
40	0.00409505	01:03.27
45	0.00118051	01:08.39
50	0.00175916	01:11.05
55	0.0015619	01:12.64
60	0.00187914	01:17.54
65	0.00125637	01:15.62
70	0.0239123	01:16.63
75	0.0029598	01:19.97
80	0.00099766	01:19.73
85	0.0133035	01:20.20
90	0.00124895	01:20.95
95	0.00119612	01:21.39
100	0.00125843	01:25.38
105	0.00153456	01:28.87
110	<b>0.000999497</b>	<b>01:28.96</b>

Pada Tabel 1 terlihat bahwa arsitektur jaringan dengan satu *hidden layer* memiliki kinerja cukup baik hal ini ditunjukkan dengan nilai error (MSE) terbaik sebesar **0.000999497** yaitu dengan jumlah neuron sebanyak 110 neuron. Secara umum jaringan dengan jumlah neuron yang semakin besar akan menghasilkan MSE yang semakin kecil.

Hasil pengamatan pelatihan kedua yang ditunjukkan oleh tabel 2 bahwa nilai pesat belajar yang cukup kecil menjamin penurunan gradient terlaksana dengan baik, namun ini berakibat bertambahnya jumlah iterasi sehingga untuk mencapai konvergensi lebih lama. Hal ini bisa dilihat untuk pesat belajar 0,6 dan momentum 0,6 dengan pesat belajar 0,9 dan momentum 0,6. Penggunaan momentum disarankan jika pencapaian nilai minimum dari suatu sistem berlangsung terlalu lama dan juga untuk mencegah terjadinya sistem terjebak dalam lokal minimum/optimum. Dalam penelitian ini hasil terbaik terjadi pada pesat belajar 0,9 dan momentum 0,6.

TABLE IV. PENGARUH MOMENTUM DAN PESAT BELAJAR DENGAN ARSITEKTUR JARINGAN SATU HIDDEN LAYER DENGAN JUMLAH NEURON 110

Mc	0.5		0.6		0.7		0.8	
	MSE	Waktu	MSE	Waktu	MSE	Waktu	MSE	Waktu
0.1	0.00131557	00:53.76	0.0012751	00:55.90	0.00155427	00:57.01	0.000999806	00:44.56
0.2	0.00103542	00:55.58	0.000999764	00:54.87	0.000999871	00:49.22	0.0120994	01:02.96
0.3	0.00145753	01:00.05	0.0245755	00:58.60	0.000999775	00:47.51	0.000999667	00:53.94
0.4	0.00138506	00:57.82	0.00163882	01:00.78	0.000999645	00:49.28	0.0241605	01:02.85
0.5	0.00141319	00:56.47	0.00116684	01:00.79	0.000999718	00:46.71	0.0119411	01:01.49
0.6	0.00108694	00:57.69	0.000999537	00:53.43	0.0123349	01:02.41	0.000999897	00:46.74
0.7	0.000999856	00:51.11	0.00107713	00:59.19	0.00162758	01:03.85	0.000999557	00:59.64

##### B. Pengujian

TABLE III. HASIL PENGUJIAN DENGAN DATA PENGUJIAN

Data	Target	Output Sistem	Error	Output Sistem (Pembulatan)
P1	1	<b>0,1456</b>	<b>0,8544</b>	<b>0</b>
P2	0	0,0000	0	<b>0</b>
P3	0	0,0000	0	<b>0</b>
P4	0	0,0000	0	<b>0</b>
P5	0	0,0000	0	<b>0</b>
P6	0	0,0001	-0,0001	<b>0</b>
P7	1	1,0000	0	<b>1</b>
P8	1	1,0000	0	<b>1</b>
P9	1	0,9885	0,0115	<b>1</b>
P10	1	1,0000	0	<b>1</b>
P11	1	1,0000	0	<b>1</b>
P12	1	1,0000	0	<b>1</b>
P13	1	1,0000	0	<b>1</b>
P14	1	1,0000	0	<b>1</b>
P15	1	<b>0,1337</b>	<b>0,8663</b>	<b>0</b>
P16	1	1,0000	0	<b>1</b>
P17	1	1,0000	0	<b>1</b>
<b>Akurasi</b>	<b>((17 - 2)/17)x100%</b>			<b>= 88,23 %</b>

Pengujian dilakukan dengan menggunakan arsitektur satu unit *hidden layer* dengan jumlah neuron 110 dengan konstanta momentum 0,6 dan pesat belajar 0,9.

Pada Tabel 3 terlihat bahwa jaringan mampu mengenali pola data pengujian dengan akurasi 15 data dari 17 data pengujian. MSE hasil pengujian sebesar 0,0870851 seperti yang ditunjukkan pada Gambar 6.

Hasil ini menunjukkan bahwa terdapat ketepatan yang baik antara output jaringan dengan target yang diharapkan. Sedangkan untuk hasil pengujian dengan data pengujian menunjukkan bahwa 2 data dari 17 data tidak sesuai dengan target. Sehingga dapat dihitung tingkat validasi dari sistem hasil penelitian ini adalah sebagai berikut [6]:

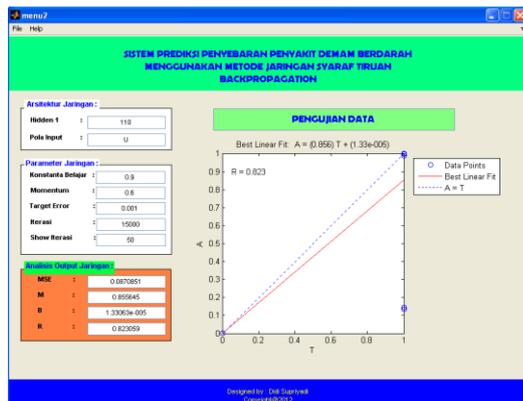
$$\text{Validasi}(\%) = \left[ \frac{\text{Jumlah\_data\_uji\_sesuai\_target}}{\text{Jumlah\_data\_uji}} \right] \times 100\%$$

$$\text{validasi}(\%) = \left[ \frac{15}{17} \right] \times 100\%$$

$$\text{Validasi} = 88,23\%$$

Keluaran jaringan akan dibandingkan dengan target data pengujian. Dari hasil perbandingan ini diketahui unjuk kerja dari sistem ini sebesar 88,23%. Sehingga apabila dilakukan pengujian berikutnya dari sekelompok data pengujian maka sebanyak 11,77% dari seluruh data pengujian tersebut akan mengalami error ataupun tidak sesuai dengan target.

0.8	0.0013107	00:58.01	0.000999879	00:53.42	0.000999521	00:46.31	0.0237904	01:03.90
0.9	0.000999792	00:50.28	<b>0.000999879</b>	<b>00:45.35</b>	0.0242226	01:01.52	0.270588	00:03.80



Gambar 6 Hasil Pengujian dengan Data Pengujian

## V. KESIMPULAN

Hasil dari penelitian ini menunjukkan bahwa dengan menggunakan parameter *inputan* seperti : suhu rata – rata ( $^{\circ}\text{C}$ ), jumlah hari hujan (hh), curah hujan (mm), kepadatan penduduk ( $/\text{km}^2$ ), ketinggian wilayah dari permukaan air laut (m), persentase ABJ (%) menunjukkan tingkat akurasi yang cukup baik yaitu sebesar 88,23% data baru yang dikenalkan ataupun diujikan sesuai dengan target sedangkan *error* sekitar 11,77% tidak dapat dikenali oleh sistem.

Error yang terdapat dalam penelitian ini dapat disebabkan oleh beberapa faktor, antara lain: Jumlah pola pelatihan yang digunakan sebanyak 85 pola data pelatihan sedangkan data yang digunakan untuk pengujian sebanyak 17 data pengujian. Kinerja jaringan mungkin akan lebih baik lagi jika jumlah data set pelatihan lebih banyak lagi. Hal ini karena semakin banyak jenis atau jumlah pola yang dilatihkan, jaringan akan semakin baik mengenali pola – pola tertentu; Jumlah unit *hidden layer* yang digunakan pada penelitian ini hanya menerapkan satu *hidden layer* dengan jumlah neuron sebanyak 110 neuron.

Untuk penelitian selanjutnya, peneliti menyarankan untuk menggunakan rentang data penelitian diambil secara

mingguan sehingga diharapkan dapat mengenali pola penyebaran penyakit demam berdarah pada wilayah tertentu.

## ACKNOWLEDGMENT

Peneliti mengucapkan terima kasih kepada Dinas Kesehatan, Badan Pusat Statistik kabupaten Brebes dan Badan Meteorologi Klimatologi dan Geofisika Kota Tegal yang telah mendukung dan membantu dalam memberikan data maupun informasi untuk tujuan penelitian ini.

## REFERENSI

- [1] Agarkar, A.M., Ghatol, A.A., 2010. FFANN Based Cost Effective Major Infant Disease Management. *International Journal of Computer Applications* (0975 - 8887)Vol. 7.
- [2] Cetiner, B.G., Sari, M., Aburas, H.M., 2009. Recognition of Dengue Disease Patterns using Artificial Neural Networks. *International Advanced Technologies Symposium 5<sup>th</sup>. Turkey*.
- [3] Chen, K., Pohan, H.T., Sinto, R., 2009. Diagnosis dan Terapi Cairan pada Demam Berdarah Dengue. *Medicinus*. Vol. 22 No.1.
- [4] Ibrahim, F., Faisal, T., Salim M., M.I., Taib, M.N., 2010. Non-Invasive Diagnosis of Risk in Dengue Patients using Bioelectrical Impedance Analysis and Artificial Neural Network. *International Federation of Medical and Biological Engineering* 48:1141-1148.
- [5] Dirjen PP & PL., 2007. Pemberantasan Sarang Nyamuk Demam Berdarah Dengue (PSN - DBD) oleh Juru Pemantau Jentik (Jumantik). Depkes RI, Jakarta.
- [6] Sandra, 2005. Aplikasi Jaringan Syaraf Tiruan untuk Pendugaan Mutu Mangga Segar Secara Non-Destruktif. *Jurnal Teknologi Pertanian*. Vol. 6 No.1 (66-72).
- [7] Hermawan, A., 2006. Jaringan Saraf Tiruan Teori dan Aplikasi. Andi, Yogyakarta.
- [8] Purnomo, M.H., Kurniawan, A., 2006. Supervised Neural Networks dan Aplikasinya. Graha Ilmu, Yogyakarta.
- [9] Siang, J.J., 2009. Jaringan Syaraf Tiruan dan Pemrogramannya menggunakan Matlab. Andi, Yogyakarta.
- [10] Whitten, J.L., dan Bentley, L.D., 2004. Metode Desain & Analisis Sistem edisi 6. Andi, Yogyakarta.
- [11] Sukamto, 2007. Studi Karakteristik Wilayah dengan Kejadian DBD di Kecamatan Cilacap Selatan Kabupaten Cilacap. Magister Kesehatan Lingkungan, Universitas Diponegoro, Semarang.