

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

Didi Supriyadi^{a,*}, Kusworo Adi^b, dan Eko Adi Sarwoko^b

^a Sekolah Tinggi Telematika, Telkom
Purwokerto, Jawa Tengah

^b Magister Sistem Informasi Program Pascasarjana
Universitas Diponegoro, Semarang

Abstract

Dengue disease is a major health problem and endemic in several countries including Indonesia. Indonesia is included in the category "A" in the stratification of DHF by WHO in 2001 which indicates the high rate of treatment in hospital and deaths from dengue. The purpose of this study was to investigate the ability of artificial neural networks Backpropagation method for information of the spread of dengue fever in a region. In this study uses six input variables which are environmental factors that influence the spread of dengue fever, include average temperature - average, rainfall, number of rainy days, the population density, sea surface height, and the percentage of larvae-free number for which data is sourced from BMKG, BPS and the Public Health Service. Network architecture applied to a multilayer network that uses an input with 6 neurons, one hidden layer and an output with the output neuron is one. From the results obtained by training the best network architecture is the number one hidden layer with the number of neurons obtained a total of 110 neurons and also the system can recognize the entire training data. The best training algorithm using a variable learning rate and momentum of 0.9 by 0.6 by the end of the training MSE 0.000999879. in the process of testing using test data obtained 17 tissue levels of approximately 88.23% accuracy. Therefore we can conclude that the network is implemented in this study when subjected to the test data other than the error rate of about 11.77%.

Keywords : Artificial Neural Networks; Backpropagation; Dengue fever

1. Pendahuluan

Demam berdarah *dengue* yang selanjutnya disebut DBD merupakan salah satu permasalahan utama yang berkembang dan endemik di benua Amerika, Eropa, Afrika, Asia, Australia, beberapa pulau di India, dan Caribbia (Ibrahim *et al.*, 2010). *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* (Ibrahim *et al.*, 2010). 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 (Chen *et al.*, 2009).

Salah satu solusi yang dapat dilakukan adalah dengan mengimplementasikan sebuah simulasi penyebaran virus *dengue* yang terjadi di Indonesia untuk memprediksi secara dini wabah *dengue*.

Tujuan penelitian ini adalah membangun sistem informasi penyebaran penyakit demam berdarah dengan metode jaringan syaraf tiruan backpropagation. Penerapan sistem informasi ini adalah untuk prediksi penyebaran penyakit demam berdarah yang disebabkan oleh nyamuk *Aedes Aegypti* yang membawa virus *dengue* (penyebab demam *dengue* maupun demam berdarah *dengue*) berdasarkan parameter-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 – parameter tersebut.

2. Kerangka Teori

Demam Berdarah *Dengue* (DBD) merupakan salah satu penyakit menular yang berbahaya dapat menimbulkan kematian dalam waktu singkat dan sering menimbulkan wabah yang disebabkan oleh virus *dengue* yang termasuk kelompok B *Arthropod Borne Virus* (*Arboviruses*) yang sekarang dikenal sebagai *genus Flavivirus*, *famili Flaviviridae*, yang ditunjukkan pada Gambar 1 dan mempunyai 4 jenis serotipe, yaitu DEN-1, DEN2, DEN-3, DEN-4. Penyakit ini ditularkan melalui gigitan nyamuk *Aedes aegypti* dan mungkin juga *Albopictus*. Kedua jenis nyamuk ini terdapat hampir di seluruh pelosok Indonesia kecuali ketinggian lebih dari 1000 meter diatas permukaan laut. Simptom *dengue* terjadi secara mendadak dan ditandai dengan demam tinggi, sakit kepala, pusing, tulang dan sendi terasa nyeri, lidahnya terasa tidak enak, dan wajahnya pucat. Pada kulit penderita terdapat bercak merah yang diakibatkan oleh gigitan nyamuk (Agarkar dan Ghatol, 2010).



Gambar 1. Virus dengue penyebab penyakit demam berdarah (Sumber :<http://wikipedia.org>)

Berdasarkan sumber dari Dirjen PP dan PL Depkes RI, 2007, pada tahun 2002 jumlah kasus sebanyak 40.377

• Alamat e-mail : didi.supriyadi.56@facebook.com

(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%).

Menurut Hairani (2009), Kriteria diagnosa penyakit DBD dapat ditegakkan dengan melihat gejala secara klinis maupun laboratorium. Gejala secara klinis penderita DBD dapat berupa: demam, pendarahan, hepatomegali/pembesaran hati, serta syok/renjatan. Sedangkan penegakan penyakit DBD berdasarkan hasil pemeriksaan laboratorium berikut : (1) *Trombositopeni* yaitu Jumlah trombosit tubuh mengalami penurunan yang drastis hingga mencapai 100.000 sel/mm^3 atau lebih rendah, (2) *Haemoconcentration* yaitu Adanya rembesan plasma karena peningkatan permeabilitas, vaskular, dimanifestasikan dengan :

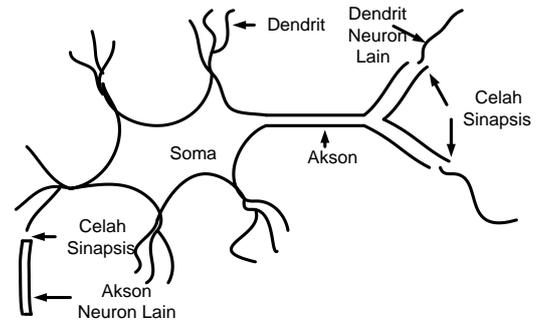
- Peningkatan hematokrit sama atau lebih besar dari 20% di atas rata – rata usia, jenis kelamin, dan populasi.
- Penurunan hematokrit setelah tindakan penggantian volume sama dengan atau lebih besar dari 20% data dasar.
- Tanda – tanda rembesan plasma seperti efusi pleural, asites, dan hipoproteinemia.

2.1. Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) adalah model komputasi dari otak manusia (Agarkar dan Ghatol, 2010). Dari pernyataan di atas terlihat jelas bahwa JST ini 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 – persoalan yang dihadapinya dan mampu belajar dari pengalaman - 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 2) yang dapat digunakan untuk memahami JST, yaitu (Siang, 2005):

- 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.
- 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.
- Akson, berfungsi sebagai saluran keluaran dari suatu neuron yang akan menyalurkan sinyal ke neuron yang lain.



Gambar 2. Neuron Tunggal

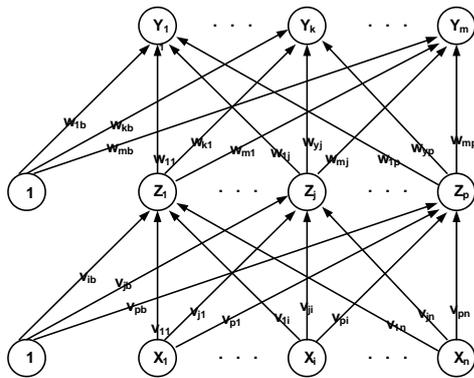
JST menyelesaikan masalah melalui proses belajar dari contoh – contoh. Biasanya JST diberikan sebuah himpunan pola pelatihan yang terdiri atas sekumpulan contoh pola. Dari contoh – contoh inilah JST melakukan proses belajar. Selama proses belajar itu pola masukan disajikan bersama – sama dengan pola keluaran yang diinginkan. Sebagai tanggapan atas pola masukan – sasaran yang disajikan tersebut, jaringan akan menyesuaikan nilai bobotnya.

JST menyimpan informasi secara terdistribusi. Sehingga akan diperoleh keuntungan pada kecepatan pengolahannya. Selain cepat, sistem terdistribusi ini juga mempunyai keuntungan yang lain, yaitu pada sifat *redundancy*. Dengan sifat ini jaringan masih tetap dapat menanggapi pola masukan dengan benar walaupun ada beberapa unit pemroses yang mengalami kerusakan (Hermawan, 2006).

1. JST *Backpropagation*

Pelatihan JST *Backpropagation* biasanya dilakukan dengan cara diawasi. Keberhasilan pelatihan sangat dipengaruhi oleh penentuan input yang tepat. 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 (Agarkar dan Ghatol, 2010).

Arsitektur JST *Backpropagation* memiliki beberapa unit yang ada dalam satu atau lebih lapis tersembunyi. Gambar 3 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 z_k).



Gambar 3. Arsitektur JST Backpropagation

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

- L0 : Inisialisasi semua 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 di atasnya.
- L4 : Hitung semua output di unit tersembunyi z_j ($j=1, 2,.., p$)

$$z_net_j = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (1)$$

$$z_j = f(z_net_j) = \frac{1}{1 + e^{-z_net_j}} \quad (2)$$

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

$$y_net_k = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad (3)$$

$$y_k = f(y_net_k) = \frac{1}{1 + e^{-y_net_k}} \quad (4)$$

FII : Backpropagation

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

$$\delta_k = (t_k - y_k) f'(y_net_k) = (t_k - y_k) y_k (1 - y_k) \quad (5)$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot lapis dibawahnya (langkah 7). Hitung suku perubahan bobot w_{kj} (yang akan dipakai nanti untuk merubah bobot w_{kj}) dengan laju percepatan α

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

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

$$\delta_net_j = \sum_{k=1}^m \delta_k w_{kj} \quad (7)$$

faktor δ unit tersembunyi:

$$\delta_j = \delta_net_j \cdot f'(z_net_j) = \delta_net_j \cdot z_j (1 - z_j) \quad (8)$$

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

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

FIII : Perubahan Bobot

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

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

Perubahan bobot garis menuju ke unit tersembunyi:

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

- L9 : Selesai

Berdasarkan algoritma di atas dapat diketahui bahwa pelatihan *backpropagation* meliputi 3 fase:

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 ($= y_k$).

Berikutnya, 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. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasikan untuk mengurangi kesalahan yang terjadi.

b. Propagasi Balik

Berdasarkan kesalahan $t_k - y_k$, dihitung faktor δ_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 . δ_k juga dipakai untuk mengubah bobot garis yang menghubungkan langsung dengan unit keluaran.

Dengan cara yang sama, dihitung δ_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 δ di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

c. Perubahan Bobot

Setelah semua faktor δ dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor δ neuron di lapis atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke lapis keluaran didasarkan atas dasar δ_k 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.

2. 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.

Sum Square Error (SSE) dihitung sebagai berikut:

- Hitung keluaran jaringan syaraf untuk masukan pertama.
- Hitung selisih antara nilai keluaran jaringan syaraf dan nilai target / yang diinginkan untuk setiap keluaran.
- 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):

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

Rumus :

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

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 suatu proses belajar jaringan syaraf tiruan ditunjukkan dengan besarnya error yang minimum. Pada kondisi inilah jaringan syaraf tiruan tersebut dapat digunakan. Ketika ada hal baru yang harus diketahui oleh jaringan syaraf tiruan maka proses belajar harus diulang kembali dengan menggunakan informasi – informasi yang lama ditambah dengan informasi – informasi yang baru. Kemampuan yang dimiliki jaringan syaraf tiruan bersifat permanent karena tidak terpengaruh oleh faktor perasaan dan firasat.

2.2. Proses Pengembangan Sistem

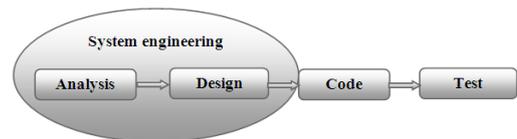
Menurut Whitten (2004), proses pengembangan sistem merupakan satu set aktivitas, metode ataupun standar proses atau langkah yang diharapkan dapat digunakan untuk pengembangan sistem. Proses pengembangan sistem pada umumnya mengikuti pendekatan pemecahan masalah. Pendekatan tersebut biasanya terdiri dari langkah pemecahan masalah yang umum:

- Mengidentifikasi masalah;
- Menganalisis dan memahami masalah;
- Mengidentifikasi persyaratan dan harapan solusi;
- Mengidentifikasi solusi alternatif dan memilih tindakan terbaik;
- Mendesain solusi yang dipilih;
- Mengimplementasi solusi yang dipilih;
- Evaluasi hasil.

Tahapan utama siklus pengembangan sistem dapat terdiri dari tahapan perencanaan sistem (*systems planning*), analisis sistem (*systems analysis*), desain sistem (*systems design*), seleksi sistem (*systems selection*),

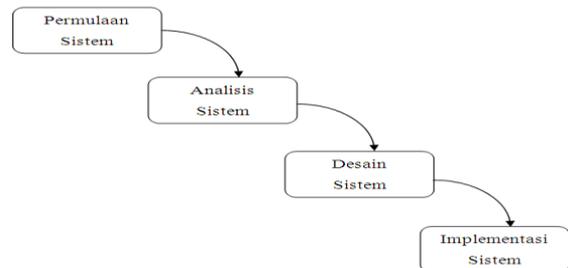
implementasi sistem (*systems implementation*), dan perawatan sistem (*systems maintenance*).

Paradigma rekayasa perangkat lunak atau yang sering dikenal sebagai model proses untuk rekayasa perangkat lunak dipilih berdasarkan sifat aplikasi dan proyeknya, metode, dan alat – alat bantu yang akan dipakai, kontrol, serta penyampaian yang dibutuhkan. Salah satu model proses rekayasa perangkat lunak adalah model sequensial linier. Model ini sering disebut sebagai siklus kehidupan klasik (*classic life cycle*) atau juga model air terjun (*waterfall model*). Model ini mengusulkan suatu pendekatan pada perangkat lunak yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem melalui beberapa tahap yaitu analisis, desain, pengkodean, pengujian, dan pemeliharaan (Pressman, 2001). Gambaran dari model proses sekuensial linier ditunjukkan pada Gambar 4.



Gambar 4. Model Sekuensial Linier (Pressman, 2001)

Menurut Whitten (2004), dan berdasarkan metode sekuensial linier di atas, beberapa tahapan proses pengembangan sistem meliputi (Gambar 5):



Gambar 5 Metode Sekuensial atau Waterfall
Sumber : (Whitten, 2004)

1. Permulaan Sistem

Permulaan sistem menentukan lingkup proyek dan rencana pemecahan masalah. Lingkup proyek mendefinisikan area bisnis yang akan ditangani oleh proyek maupun tujuan – tujuan yang akan dicapai.

2. Analisis Sistem

Analisis sistem ini dimaksudkan untuk memberikan pemahaman yang lebih mendalam kepada tim proyek tentang masalah dan kebutuhan yang memicu proyek tersebut. Analisis sistem memerlukan kerja dengan pengguna sistem untuk secara jelas mendefinisikan kebutuhan bisnis dan harapan untuk setiap sistem baru.

3. Desain Sistem

Kita dapat melakukan proses desain sistem untuk memberikan pemahaman tentang kebutuhan sistem informasi. Pada saat mendesain sistem, kita dituntut untuk dapat mengeksplorasi suatu solusi teknis yang kemudian dipilih untuk dikembangkan menjadi solusi akhir dari permasalahan pada penelitian. Sistem desain ini

melibatkan program, user interface, serta jaringan untuk sistem informasi.

4. Implementasi Sistem

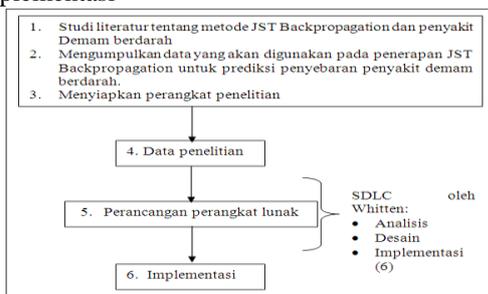
Langkah terakhir dalam proses pengembangan sistem adalah implementasi sistem. Implementasi sistem membangun sistem informasi baru dan menempatkannya ke dalam operasi. Selama implementasi sistem proses yang dilakukan adalah memasang dan menguji perangkat keras maupun perangkat lunak sistem baru.

3. Metodologi

3.1. Desain Penelitian

Pada penelitian ini tahapan – tahapan desain penelitian yang dilakukan antara lain (Gambar 6):

1. Studi literatur untuk mempelajari tentang metode yang akan digunakan dalam penelitian yaitu metode jaringan syaraf tiruan backpropagation serta studi tentang kasus yang akan diteliti yaitu kasus prediksi penyebaran penyakit demam berdarah.
2. Mengumpulkan data yang akan digunakan pada penerapan JST Backpropagation untuk prediksi penyebaran penyakit demam berdarah.
3. Menyiapkan perangkat penelitian.
4. Data penelitian
5. Perancangan perangkat lunak untuk prediksi penyebaran penyakit demam berdarah menggunakan metode JST Backpropagation
6. Implementasi



Gambar 6. Desain penelitian

3.2. Metode Penelitian

Metodologi yang diterapkan dalam penelitian ini meliputi metode pengumpulan data dan metode pengembangan sistem.

1. Metode pengumpulan data

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

- a. Membuat rancangan data input dan data output sebagai data pelatihan maupun data pengujian
Berdasarkan keterangan ahli tersebut, data yang digunakan dalam penelitian ini meliputi: data suhu rata – rata ($^{\circ}\text{C}$), data curah hujan (mm), data jumlah hari hujan (hh), data Kepadatan penduduk (km^2), data ketinggian daratan dari permukaan air laut (m), serta data persentase ABJ (%).
Data-data tersebut diperoleh dari berbagai sumber dari pemerintah Kabupaten Brebes, untuk data suhu rata – rata diperoleh dari Badan Meteorologi Klimatologi dan

Geofisika (BMKG) Kabupaten Brebes yang masih menginduk dengan BMKG Kota Tegal, sedangkan data curah hujan, jumlah hari hujan, kepadatan penduduk dan ketinggian wilayah diperoleh dari Badan Pusat Statistik (BPS) Kabupaten Brebes dan untuk data persentase Angka Bebas Jentik (% ABJ) dari Dinas Kesehatan Kabupaten Brebes. Selain enam parameter tersebut yang digunakan sebagai variabel input dari sistem yang akan dikembangkan, dalam penelitian ini juga mengambil data kasus DBD yang digunakan sebagai target ataupun output dari sistem yang diperoleh dari Dinas Kesehatan Kabupaten Brebes. Pasangan input dan target yang digunakan dalam penelitian ini diperlihatkan dalam Gambar 7.

Prediksi penyebaran penyakit DBD dengan menggunakan jaringan syaraf tiruan akan semakin baik apabila data historis yang digunakan dalam proses pelatihan semakin banyak. Data primer yang digunakan dalam penelitian ini adalah data yang diambil secara *time series*, sehingga ada 102 pasangan data yang telah diperoleh merupakan data nilai tahunan dari enam parameter tersebut yang meliputi 17 kecamatan yang ada di Kabupaten Brebes dari tahun 2005 – 2010.

| | | |
|-------------------------------|---|--------|
| Suhu Rata – Rata | } | Input |
| Curah Hujan | | |
| Jumlah Hari Hujan | | |
| Kepadatan Penduduk | | |
| Ketinggian Permukaan Air Laut | | |
| % Angka Bebas Jentik | | |
| Kondisi DBD / Bebas | } | Target |

Gambar 7. Pasangan Input dan Target Sistem

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

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

2. Metode pengembangan sistem

Pada penelitian ini peneliti menggunakan teknik pengembangan sistem model yang dikembangkan oleh Whitten yang dapat dijabarkan sebagai berikut:

- a. Reayasa dan Pemodelan Sistem
Untuk menggambarkan sistem prediksi penyebaran penyakit DBD dengan metode JST *Backpropagation* yang akan dikembangkan, maka perlu dibuat terlebih dahulu 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

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

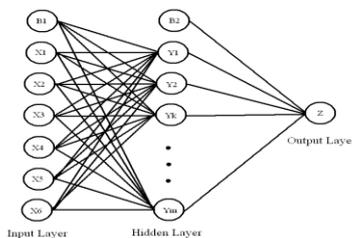
Rancangan jumlah neuron pada JST *Backpropagation* untuk sistem prediksi penyebaran penyakit DBD:

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 berdasarkan keterangan dari ahli epidemiologi dan data yang diperoleh di lokasi penelitian. Data ini dimasukkan ke dalam JST dalam bentuk bilangan real yang berkisar antara 0 s.d 1.

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.

Pengembangan sistem prediksi penyebaran penyakit DBD dengan metode JST *Backpropagation* dimulai dari perancangan arsitektur jaringan seperti pada Gambar 8.



Gambar 8. Arsitektur JST untuk Sistem Prediksi Penyebaran Penyakit DBD

Arsitektur jaringan dipilih dengan pendekatan konstruktif, yaitu dengan suatu jaringan yang kecil dengan satu *hidden layer* kemudian mengembangkan jumlah unit tersembunyi serta bobot tambahan sampai didapatkan penyelesaian yang dikehendaki.

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 yang disimpan sebagai matriks dengan nama *P2.dat*.

Data tersebut kemudian diolah ke dalam suatu *range* tertentu yang disebut *preprocessing* atau 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 adalah berdasarkan rumus (Purnomo, M.H., 2006):

$$A_i' = \frac{A_i - A_{\min}}{A_{\max} - A_{\min}}$$

(14)

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

Setelah dinormalisasi maka proses selanjutnya adalah proses komputasi *backpropagation* dengan menerapkan semua rumus dan algoritma dari *backpropagation* yang telah dijelaskan (2.2).

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 ataupun fase yang telah dijelaskan.

Langkah nomor 3 akan terus diulangi sampai kondisi berhenti terpenuhi sehingga akan memerlukan banyak sekali iterasi sebelum proses pelatihan berhenti.

Setelah membangun JST *Backpropagation* dan menentukan parameter jaringan langkah komputasi selanjutnya adalah melakukan proses pelatihan JST.

Pelatihan dilakukan dengan variasi parameter jaringan yaitu jumlah neuron pada masing-masing *hidden layer*, pesat belajar, dan konstanta momentum. Setiap variasi parameter diamati dengan menghitung MSE dan lama iterasinya.

- 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. Untuk menentukan jumlah neuron pada *hidden layer*, 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 1 dilakukan dengan konstanta pesat belajar 0,1, konstanta momentum 0,7, kinerja tujuan (target *error*) 0,001, maksimum iterasi 10000 *epoch* dan dengan algoritma pelatihan *gradient descent* dengan momentum dan pesat belajar (traingdx).

- Pelatihan 2, Mencari Momentum dan Laju Pembelajaran Terbaik

Pada langkah ini akan diamati variasi momentum dan pesat belajar terhadap unjuk kerja jaringan yang dinyatakan dengan besar kecilnya MSE selama proses pelatihan dan lamanya waktu iterasi. Tujuan dari langkah ini adalah untuk memperoleh nilai momentum dan pesat

belajar yang optimum. Dengan memakai arsitektur jaringan terbaik pada pelatihan 1 maka akan diamati pengaruh momentum dan pesat belajar terhadap unjuk kerja jaringan yaitu pada proses pelatihan.

Pada pelatihan 2, pengamatan dilakukan secara konstruktif dengan konstanta pesat belajar 0,1 sampai dengan 0,9 dan juga dengan variasi momentum mulai dari 0,5 sampai dengan 0,8. Kinerja tujuan (target error) yang diberikan adalah 0,001, maksimum iterasinya 15000 epoch dan algoritma pelatihnnya *gradient descent* dengan momentum dan pesat belajar (traingdx).

• Pengujian

Jaringan yang telah dilatih dan mencapai hasil yang dikehendaki perlu diuji untuk mengetahui kemampuannya pada saat mempelajari data latih yang diberikan. 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.

4. Hasil dan Pembahasan

4.1. Pelatihan

Proses pelatihan merupakan proses yang sangat penting, karena akan menentukan berhasil tidaknya sebuah sistem jaringan yang akan digunakan dalam mengenali pola masukan dan keluaran. Pada bagian ini jaringan dilatih dengan menggunakan pola pasangan masukan dan keluaran untuk mendapatkan bobot koneksi yang tepat antar lapisan. Secara umum semakin banyak pola pelatihan yang dipakai, hasilnya akan semakin baik.

Dari hasil pelatihan 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 pada *hidden layer* sebanyak 110 neuron. Secara umum jaringan dengan jumlah neuron yang semakin besar akan menghasilkan MSE yang semakin kecil. Akan tetapi jaringan dengan jumlah neuron yang semakin besar, waktu komputasinya juga semakin lama.

Parameter pesat belajar sangat berpengaruh terhadap efektivitas dan kecepatan dari sistem untuk mencapai konvergensi/nilai minimum dari suatu pelatihan. Dari data hasil pengamatan pada pelatihan kedua yang ditunjukkan oleh Tabel 2 bahwa untuk nilai pesat belajar yang cukup kecil menjamin penurunan gradient terlaksana dengan baik, namun ini berakibat bertambahnya jumlah iterasi sehingga untuk mencapai konvergensi berlangsung lebih lama. Hal ini bisa dilihat untuk pesat belajar 0,6 dengan momentum 0,6 akan membutuhkan waktu iterasi yang lebih lama untuk mencapai nilai minimum yaitu selama 00:53.43 menit dibandingkan dengan pesat belajar 0,9 dengan momentum 0,6 membutuhkan waktu lebih sedikit untuk mencapai nilai minimum yaitu selama 00:45.35 menit.

Konstanta momentum bertujuan untuk mempercepat konvergensi dari algoritma *backpropagation*. Penggunaan konstanta momentum disarankan jika pencapaian nilai

mínimum dari suatu sistem berlangsung terlalu lama dan juga untuk mencegah terjadinya sistem terjebak dalam lokal mínimum ataupun lokal optimum. Dalam penelitian ini hasil terbaik dari variasi pesat belajar dan momentum terjadi pada pesat belajar 0,9 dan momentum 0,6 yang menghasilkan MSE sebesar 0.000999879/0.001 dengan waktu iterasi selama 00:45.35 menit.

Tabel 1. Hasil pelatihan dengan satu hidden layer untuk jumlah neuron hidden layer yang divariasi

| Jumlah Neuron (nL1) | MSE | Time |
|---------------------|-------------|----------|
| 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 | 0.000999766 | 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 | 0.000999497 | 01:28.96 |

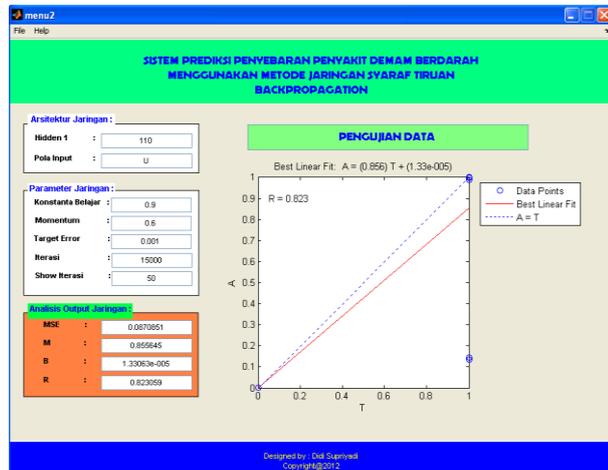
Tabel 2. Pengaruh Momentum dan Pesat Belajar dengan Arsitektur Jaringan Satu Hidden Layer dengan Jumlah Neuron 110

| Mc Lr | 0.5 | | 0.6 | |
|----------|-------------|----------|-------------|----------|
| | MSE | Time | MSE | Time |
| 0.1 | 0.00131557 | 00:53.76 | 0.0012751 | 00:55.90 |
| 0.2 | 0.00103542 | 00:55.58 | 0.000999764 | 00:54.87 |
| 0.3 | 0.00145753 | 01:00.05 | 0.0245755 | 00:58.60 |
| 0.4 | 0.00138506 | 00:57.82 | 0.00163882 | 01:00.78 |
| 0.5 | 0.00141319 | 00:56.47 | 0.00116684 | 01:00.79 |
| 0.6 | 0.00108694 | 00:57.69 | 0.000999537 | 00:53.43 |
| 0.7 | 0.000999856 | 00:51.11 | 0.00107713 | 00:59.19 |
| 0.8 | 0.0013107 | 00:58.01 | 0.000999879 | 00:53.42 |
| 0.9 | 0.000999792 | 00:50.28 | 0.000999879 | 00:45.35 |
| 0.1 | 0.00155427 | 00:57.01 | 0.000999806 | 00:44.56 |
| 0.2 | 0.000999871 | 00:49.22 | 0.0120994 | 01:02.96 |
| 0.3 | 0.000999775 | 00:47.51 | 0.000999667 | 00:53.94 |
| 0.4 | 0.000999645 | 00:49.28 | 0.0241605 | 01:02.85 |
| 0.5 | 0.000999718 | 00:46.71 | 0.0119411 | 01:01.49 |
| 0.6 | 0.0123349 | 01:02.41 | 0.000999897 | 00:46.74 |
| 0.7 | 0.00162758 | 01:03.85 | 0.000999557 | 00:59.64 |
| 0.8 | 0.000999521 | 00:46.31 | 0.0237904 | 01:03.90 |
| 0.9 | 0.0242226 | 01:01.52 | 0.270588 | 00:03.80 |

4.2. Pengujian

Pengujian dilakukan dengan menggunakan data yang belum pernah dilatih sebelumnya. Tujuan dari pengujian ini adalah untuk mengetahui seberapa jauh sistem dapat mengenali pola yang telah dilatihkan. Pengujian dilakukan dengan menggunakan arsitektur satu unit *hidden layer* dengan jumlah neuron 110 dengan konstanta momentum 0,6 dan pesat belajar 0,9. Hasil pengujian dengan data pengujian disajikan dalam Tabel 3.

Pada Tabel 3 terlihat bahwa jaringan sudah mampu mengenali pola data pengujian dengan akurasi 15 data dari 17 data pengujian. MSE hasil pengujian sebesar 0,0870851 seperti yang ditunjukkan pada Gambar 9.



Gambar 9. Hasil Pengujian dengan Data Pengujian

Tabel 3. Hasil pengujian dengan data pengujian

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

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 :

$$Validasi(\%) = \left[\frac{Jumlah_data_uji_sesuai_target}{Jumlah_data_uji} \right] \times 100\%$$

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

$$Validasi = 88,23\%$$

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.

Pengujian selanjutnya adalah mengamati kemampuan sistem dalam memberikan fungsi sebagai penunjang keputusan. Pengujian dilakukan dengan cara mengambil salah satu data uji yang memiliki nilai kondisi real maupun kondisi keluaran sistem adalah kondisi 1 ataupun kondisi terjadi kasus DBD yang kemudian mengubah – ubah nilai persentase Angka bebas jentik sehingga diperoleh hasil keluaran sistem adalah kondisi 0 atau kondisi tidak terjadi kasus DBD.

Pengujian ini mengambil sampel data uji kecamatan K9. Salah satu proses pengujian ini ditunjukkan Gambar 10 dengan tabel pengamatan ditunjukkan pada Tabel 4.

Tabel 4. Hasil Pengamatan dengan mengubah-ubah Nilai Variabel %ABJ

| Data | Kondisi Real | Keterangan Kondisi | %ABJ | Output Sistem | Keterangan Sistem |
|-------|--------------|--------------------|------|---------------|-------------------|
| K9=P9 | 1 | DBD | 75 | 0 | Bebas |
| | 1 | DBD | 80 | 0 | Bebas |
| | 1 | DBD | 85 | 0 | Bebas |
| | 1 | DBD | 90 | 0 | Bebas |
| | 1 | DBD | 95 | 0 | Bebas |
| 1 | DBD | 100 | 0 | Bebas | |

Berdasarkan Tabel 3 diketahui bahwa pada saat pengujian data uji P9 bernilai kondisi 1 atau terjadi kasus DBD. Selanjutnya dari Gambar 10 dan Tabel 4 dapat dianalisis bahwa kondisi suatu wilayah terhadap penyebaran penyakit DBD dapat berubah dari kondisi terjadi kasus DBD ke kondisi tidak terjadi kasus DBD ataupun sebaliknya. Hal ini dapat dilihat dari tabel 4 (data uji K9) bahwa dengan mengubah nilai variabel %ABJ diubah sampai dengan 100% maka keluaran sistem akan berubah menjadi 0 ataupun tidak terjadi kasus DBD yang diujicobakan pada saat nilai %ABJ 75% sampai 100%. Hal ini dapat digunakan sebagai penunjang keputusan dalam rangka penatalaksanaan penyebaran penyakit DBD pada wilayah tersebut bahwa untuk mencegah terjadinya penyebaran kasus DBD salah satu cara yang dapat dilakukan adalah dengan meningkatkan persentase angka bebas jentik.

The screenshot shows a web application interface with the following sections:

- FAKTOR YANG MEMPENGARUHI :** A table of input variables:

| | | |
|--------------------|---|------|
| Suhu Rata - Rata | = | 27,6 |
| Jumlah Hari Hujan | = | 210 |
| Curah Hujan | = | 3511 |
| Kepadatan Penduduk | = | 648 |
| Ketinggian Wilayah | = | 22 |
| % ABJ | = | 85 |
- INFORMASI** section with a text box containing '0' and the text 'Tidak Ada Kasus' below it.
- PREDIKSI TINGKAT ENDEMISITAS :**
 - Hasil Prediksi Tahun Ini = 0
 - Kasus 1 Tahun Lalu =
 - Kasus 2 Tahun Lalu =
- Buttons: **RESET** and **KELUAR**.
- Footer: Designed by : Didit Supriyadi, Copyright©2012.

Gambar 10. Proses Pengujian dengan mengubah Nilai Variabel %ABJ

5. 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. Hal ini tidak sesuai dengan penelitian yang dilakukan oleh peneliti sebelumnya yaitu Cetiner (2009), bahwa validasi menjadi sangat baik jika data yang digunakan untuk pelatihan jumlahnya lebih kecil daripada jumlah data pengujian. Hal ini berarti 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. Menurut Hermawan (2006), jaringan dengan lapisan banyak dapat menyelesaikan permasalahan yang lebih sulit dan kompleks jaringan daripada dengan lapisan tunggal, akan tetapi proses pembelajarannya lebih rumit.

Daftar Pustaka

- Agarkar, A.M., Ghatol, A.A., 2010. FFANN Based Cost Effective Major Infant Disease Management. International Journal of Computer Applications (0975 - 8887), Vol. 7.
- Chen, K., Pohan, H.T., Sinto, R., 2009. Diagnosis dan Terapi Cairan pada Demam Berdarah Dengue. Medicinus. Vol. 22 No.1.
- 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.
- Dirjen PP dan PL., 2007. Pemberantasan Sarang Nyamuk Demam Berdarah Dengue (PSN - DBD) oleh Juru Pemantau Jentik (Jumantik). Depkes RI, Jakarta.
- Hermawan, A., 2006. Jaringan Saraf Tiruan Teori dan Aplikasi. Andi, Yogyakarta.
- Pressman, R.S., 2001. Software Engineering A Practitioner's Approach 5th edition. McGraw-Hill. New York.
- Purnomo, M.H., Kurniawan, A., 2006. Supervised Neural Networks dan Aplikasinya. Graha Ilmu, Yogyakarta.
- Siang, J.J., 2009. Jaringan Syaraf Tiruan dan Pemrogramannya menggunakan Matlab. Andi, Yogyakarta.
- Whitten, J.L., dan Bentley, L.D., 2004. Metode Desain & Analisis Sistem edisi 6. Andi, Yogyakarta.
- Hairani, L.K., 2009. Gambaran Epidemiologi. FKM UI, Jakarta.
- Sukanto, 2007. Studi Karakteristik Wilayah dengan Kejadian DBD di Kecamatan Cilacap Selatan Kabupaten Cilacap. Magister Kesehatan Lingkungan, Universitas Diponegoro, Semarang.