

**SISTEM INFORMASI PENYEBARAN PENYAKIT DEMAM BERDARAH
MENGUNAKAN METODE JARINGAN SYARAF TIRUAN
*BACKPROPAGATION***

**Tesis
Untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-2
Program Studi Magister Sistem Informasi**



**Oleh :
Didi Supriyadi
24010410400013
PROGRAM PASCASARJANA
UNIVERSITAS DIPONEGORO
SEMARANG**

2012

ABSTRAK

Penyakit Demam Berdarah merupakan salah satu permasalahan kesehatan utama dan endemik di beberapa Negara termasuk di Indonesia. 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 dari penelitian ini adalah untuk menginvestigasi kemampuan metode Jaringan syaraf tiruan *Backpropagation* untuk informasi penyebaran penyakit demam berdarah pada suatu wilayah. Dalam penelitian ini menggunakan enam variabel input yang merupakan faktor lingkungan yang mempengaruhi penyebaran penyakit demam berdarah, meliputi suhu rata – rata, curah hujan, jumlah hari hujan, kepadatan penduduk, ketinggian permukaan air laut, dan persentase angka bebas jentik yang datanya bersumber dari BMKG, BPS dan Dinas Kesehatan. 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 pula sistem dapat mengenali seluruh data pelatihan. Algoritma pelatihan terbaik dengan menggunakan variabel pesat belajar sebesar 0,9 dan momentum sebesar 0,6 dengan MSE akhir pelatihan 0.000999879. pada proses pengujian menggunakan 17 data pengujian diperoleh tingkat akurasi jaringan sekitar 88,23%. Sehingga dapat disimpulkan bahwa jaringan yang diimplementasikan pada penelitian ini ketika dikenai data pengujian yang lain maka tingkat kesalahannya sekitar 11,77%.

Keywords : Jaringan syaraf tiruan; *Backpropagation*; Demam berdarah

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.*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penyakit 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). Penyakit DBD adalah penyakit infeksi yang ditularkan dari orang ke orang oleh nyamuk, yang disebut nyamuk *Aedes Aegypti* (Cetiner et al, 2009). Waktu infeksi dari gigitan nyamuk *Aedes Aegypti* betina pada manusia sehingga dapat menularkan virus *dengue* masa inkubasinya hanya sekitar 8 – 10 hari (Cetiner et al, 2009).

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). Sampai saat ini infeksi virus *dengue* tetap menjadi masalah kesehatan masyarakat dan merupakan penyakit endemis hampir di seluruh provinsi di Indonesia. 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).

Pada tahun 2009 jumlah kasus DBD di Indonesia sebanyak 158.912 (AI > 55 kasus/100.000 penduduk dengan CFR di atas 1%) (Depkes, 2009). Penyakit DBD ditandai dengan demam tinggi, sakit kepala, nyeri pada beberapa bagian ruam tubuh, *lymphodenopathy*, dan *leucopenia* (Cetiner et al, 2009). Penyebab dari permasalahan ini tidak jelas akan tetapi kemungkinan disebabkan salah satunya oleh peningkatan populasi, urbanisasi, travel, dan iklim yang berubah (WHO 1999) (Hales dan Hearnden, 1999). Keterlambatan dalam mengidentifikasi masalah ini dapat menyebabkan keterlambatan dalam mengendalikan dan akhirnya akan dapat mengakibatkan kejadian ataupun wabah yang di luar kendali, sehingga demam *dengue* maupun demam berdarah *dengue* perlu dikelola secara optimum karena masa inkubasi maupun perkembangan virus *dengue* terjadi dalam waktu yang sangat singkat.

Salah satu solusi yang dapat dilakukan adalah dengan mengimplan sebuah simulasi penyebaran virus *dengue* yang terjadi di Indonesia untuk menginformasikan secara dini wabah *dengue*. Tujuan dari penelitian ini adalah untuk menginvestigasi kemungkinan penerapan Jaringan Syaraf Tiruan (JST) sebagai teknik untuk informasi penyebaran penyakit

demam berdarah yang disebabkan oleh nyamuk *Aedes Aegypti* yang membawa virus *dengue* (penyebab demam *dengue* maupun demam berdarah *dengue*) berdasarkan parameter : suhu rata – rata ($^{\circ}\text{C}$), kepadatan penduduk (populasi)/ km^2 , curah hujan (mm), jumlah hari hujan (hh), %ABJ (%) dan ketinggian dari permukaan air laut (m) dan laporan kasus *dengue* sebagai koresponden keluaran dari parameter – parameter tersebut.

Pada tesis ini JST *backpropagation* akan dicoba diterapkan untuk menginformasikan penyebaran suatu penyakit yang berkembang di masyarakat yaitu Demam berdarah berdasarkan faktor lingkungan. JST *backpropagation* merupakan topologi yang cukup populer dan paling banyak dipakai untuk berbagai aplikasi terutama pengenalan pola. JST *backpropagation* adalah jenis *supervised learning* dimana *output* dari jaringan dibandingkan dengan target yang diharapkan sehingga diperoleh *error output*, kemudian *error* ini dipropagasikan balik untuk memperbaiki bobot jaringan dalam rangka meminimasi *error*. Pada sistem informasi penyebaran penyakit demam berdarah berbasis JST, keberhasilan tergantung pada data-data yang telah diberikan pada fase pelatihan.

1.2 Perumusan Masalah

Peristiwa bertambahnya penderita atau kematian yang disebabkan oleh suatu penyakit di suatu wilayah tertentu, kadang – kadang dapat merupakan kejadian yang mengejutkan banyak orang. Secara umum kejadian ini sering disebut Kejadian Luar Biasa (KLB). Wabah DBD masih tergolong kejadian luar biasa (KLB) di Indonesia saat ini. Untuk mencegahnya maka dibutuhkan sebuah sistem yang dapat digunakan untuk membantu mengetahui penyebaran penyakit *dengue* yang disebabkan oleh nyamuk *Aedes Aegypti* betina berdasarkan parameter : suhu rata – rata ($^{\circ}\text{C}$), curah hujan (mm), jumlah hari hujan (hh), kepadatan penduduk (populasi)/ km^2 , %ABJ (angka bebas jentik) (%), dan ketinggian daratan dari permukaan air laut (m) dan laporan kasus *dengue* sebagai koresponden keluaran dari parameter – parameter tersebut.

JST telah dikembangkan untuk mensimulasikan kemampuan proses belajar dari otak manusia (Efendigil et al, 2008). JST telah banyak dikembangkan dan digunakan untuk keperluan prediksi, clustering, klasifikasi, dan penanda pola yang tidak normal (Haykin, 1994). Caranya adalah dengan melakukan peniruan terhadap aktivitas – aktivitas yang terjadi di dalam sebuah jaringan syaraf biologis. Ketika manusia berpikir, aktivitas yang terjadi adalah aktivitas mengingat, memahami, menyimpan, dan memanggil kembali apa yang pernah dipelajari oleh otak. Ide dari jaringan syaraf biologis yang diambil pada JST adalah adanya elemen – elemen pemrosesan pada JST yang saling terhubung dan beroperasi secara

paralel (Puspitaningrum, 2006). Maka perumusan masalah pada penelitian ini adalah “Apakah metode JST *Backpropagation* dapat digunakan sebagai metode untuk menginformasikan pola penyebaran penyakit demam berdarah di Indonesia berdasarkan parameter – parameter yang telah ditentukan dan berdasarkan keterangan ahli di Kabupaten Brebes?

1.3 Batasan Masalah

Pengembangan sistem informasi penyebaran penyakit demam berdarah ini dibangun menggunakan metode JST *Backpropagation* untuk mencapai minimum *error* dengan variasi laju pembelajaran dan momentum untuk modifikasi JST *Backpropagation*.

1.4 Keaslian Penelitian

Penelitian terkait tentang demam *dengue* maupun demam berdarah *dengue* dan penerapan metode JST *Backpropagation* yang sudah pernah dilakukan antara lain sebagai berikut :

1. Cetiner, B. Gultekin, et al (2009) dengan judul penelitian : *Recognition of Dengue Disease Patterns using Artificial Neural Networks*.

Penelitian bertujuan untuk menginvestigasi kemungkinan penerapan teknik JST *Backpropagation* untuk pengenalan pola penyebaran penyakit demam berdarah berdasarkan parameter : suhu rata – rata, kelembaban rata - rata dan curah hujan. Data yang digunakan bersumber dari *Singaporean National Environment Agency* (NEA) yang diset secara mingguan dari bulan Januari 2001 – April 2007 dengan jumlah data sebanyak 330 set data.

2. Ibrahim, et al (2010) dalam penelitian yang berjudul : *Non – invasive diagnosis of risk in dengue patients using bioelectrical impedance analysis and artificial neural network*.

Penelitian ini mengembangkan sistem untuk mendiagnosis dan klasifikasi resiko pasien demam berdarah secara dini menggunakan BIA dan JST. Jumlah data yang digunakan sebanyak 223 subyek sehat dan 207 data pasien demam berdarah. Parameter yang digunakan untuk proses pelatihan dan pengujian adalah lamanya demam, reaktansi, jenis kelamin, dan kualifikasi group resiko. Arsitektur JST yang digunakan adalah JST *Backpropagation* dengan algoritma *gradient descent* dengan momentum, fungsi aktivasi yang digunakan adalah logsig, dan arsitekturnya adalah 3 *input*, 6 *neuron* hidden layer, dan 1 *output*.

Dari penelitian sebelumnya, perbedaan penelitian ini pada beberapa parameter tambahan yang digunakan sebagai variabel *input*. Berdasarkan keterangan dari ahli epidemiologi, variabel tersebut adalah : curah hujan (mm), jumlah hari hujan (hh), ketinggian daratan dari permukaan air laut (m), dan persentase angka bebas jentik (%ABJ) (%), yang pada penelitian sebelumnya belum dilakukan.

1.5 Manfaat Hasil Penelitian

Penelitian ini diharapkan dapat memberikan manfaat terhadap ilmu pengetahuan pada bidang teknologi informasi dengan menerapkan metode *JST Backpropagation* untuk informasi penyebaran penyakit demam berdarah.

1.6 Tujuan Penelitian

Mengimplementasikan metode *JST Backpropagation* sebagai alat untuk menginformasikan penyebaran penyakit demam berdarah *dengue* di Kabupaten Brebes.

1.7 Ruang Lingkup

a) Ruang Lingkup Keilmuan

Menerapkan metode *JST Backpropagation* pada bidang kesehatan untuk menginformasikan penyebaran penyakit Demam Berdarah.

b) Ruang Lingkup Materi

Materi dibatasi pada penerapan metode *JST Backpropagation* untuk informasi penyebaran penyakit demam berdarah. Aplikasi ini nantinya diharapkan akan mampu mendeteksi kondisi wilayah apakah memiliki resiko terjangkit penyakit demam berdarah atau tidak. Hal ini dikarenakan beberapa alasan yaitu ketika peneliti berkonsultasi dengan staf ahli Pengendali Penyakit Dinas Kesehatan Kabupaten Brebes yaitu Awaludin SKM, MKes. yang menangani langsung pengendalian penyakit termasuk di dalamnya menangani penyebaran penyakit demam berdarah menyatakan bahwa untuk dapat mengklasifikasikan DBD sampai dengan tipe – tipe DBD pada suatu wilayah dilakukan dengan melihat data kasus 2 tahun sebelumnya apakah pada 2 tahun sebelumnya terdapat kasus DBD.

c) Ruang Lingkup Variabel

Variabel yang digunakan untuk menginformasikan penyebaran penyakit DBD di Kabupaten Brebes didasarkan pada keterangan ahli yang diperoleh yaitu karakteristik faktor wilayah yang meliputi : suhu rata – rata ($^{\circ}\text{C}$), curah hujan (mm), jumlah hari hujan

(hh), kepadatan penduduk ($/\text{km}^2$), ketinggian daratan dari permukaan air laut (m), dan persentase Angka Bebas Jentik (%ABJ) (%). Data variabel yang akan digunakan dalam penelitian ini merupakan data yang di koleksi secara tahunan dan bersumber dari BMKG kota Tegal, BPS dan Dinas Kesehatan Kabupaten Brebes.

d) Ruang Lingkup Lokasi

Lokasi penelitian ini dilaksanakan di Kabupaten Brebes Provinsi Jawa Tengah.

e) Ruang Lingkup Waktu

Waktu yang digunakan untuk melaksanakan penelitian ini yakni pada bulan November 2011 sampai dengan selesai.

BAB II TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

JST tersusun atas elemen-elemen sederhana yang beroperasi secara paralel. Elemen-elemen ini diilhami oleh sistem syaraf biologis sehingga dapat menirukan kerja otak manusia. JST yang telah dilatih dapat melakukan fungsi-fungsi kompleks di berbagai bidang, antara lain identifikasi, pengenalan pola, klasifikasi, restorasi citra, termasuk juga untuk sistem kendali (Demuth dan Beale, 1994).

JST dapat menyelesaikan berbagai masalah teknik pada beberapa area mulai dari komputer *vision* hingga peramalan bisnis berdasarkan ketersediaan data. Kemampuannya untuk belajar dari contoh – contoh yang diberikan, menjadikannya sebagai perangkat yang baik karena tidak perlu model matematis dari tiap kasus individu (Cetiner et al, 2009). JST tersusun atas sejumlah *neuron* yang saling terhubung yang terbagi ke dalam beberapa lapisan yaitu lapisan *input*, *hidden*, dan *output*. Pada lapisan *input*, suatu JST terdiri atas beberapa *neuron*, hal ini merepresentasikan parameter dari suatu kasus yang terhubung ke lapisan *hidden*, dan lapisan *hidden* saling terhubung ke lapisan *output* (Efendigil et al, 2008).

Pada penelitian sebelumnya (Cetiner et al, 2009), telah melakukan penelitian yang bertujuan untuk menginvestigasi kemungkinan penerapan teknik JST untuk pengenalan pola penyebaran *dengue* berdasarkan 3 parameter utama yaitu : suhu rata – rata, kelembaban rata – rata, dan curah hujan dengan laporan kasus *dengue* sebagai keluaran dari sistem yang dikembangkan. Dalam penelitian tersebut menggunakan data kasus *dengue* yang terjadi dari tahun 2001 – 2007 yang terdiri atas 330 set data yang bersumber dari *Singaporean National Environment Agency* (NEA).

Dalam penelitiannya Cetiner membagi 330 set data menjadi dua bagian yaitu sebanyak 104 set data digunakan untuk pelatihan dan 226 set data untuk pengujian. Rasio perbandingan data tersebut adalah 104/226 atau sekitar 0,46. Dengan demikian validasi menjadi sangat baik jika pelatihan dan pengujian yang dilakukannya sukses karena data yang digunakan untuk pelatihan lebih kecil daripada data pengujian.

Dalam penelitian yang dilakukan Cetiner, JST melaksanakan pelatihan dan pengujian untuk pengenalan pola penyebaran penyakit *dengue* dengan menggunakan 330 set data. Korelasi tertinggi yang dicapai adalah 0,85 pada saat pelatihan. Satu *hidden layer* telah ditentukan secara optimum dengan menggunakan 4 elemen pemroses (jumlah *neuron hidden*

layer) berdasarkan *trial and error*. Akan tetapi, pada saat sistem digunakan untuk memprediksi kejadian yang terjadi pada tahun 2005, korelasi pengujiannya turun menjadi 0,70 dari yang sebelumnya 0,76. Hal ini dikarenakan pada minggu ke 34 – 38 tahun 2005 akurasi sangat rendah. Hal ini menunjukkan bahwa dalam periode tersebut terdapat beberapa kejadian yang tidak dapat dijelaskan oleh ketiga parameter yang telah digunakan. Sehingga untuk studi selanjutnya perlu ditambahkan parameter lain untuk memperoleh tingkat akurasi maupun korelasi prediksi yang lebih baik lagi (Cetiner et al, 2009).

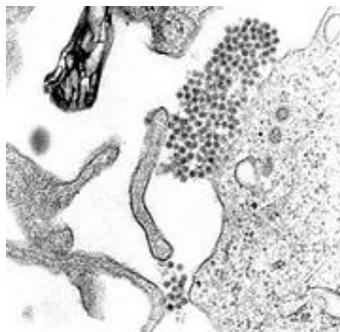
Dalam penelitian (Ibrahim et al, 2010), untuk mendiagnosa dan mengklasifikasi resiko dini penderita demam *dengue* menggunakan metode JST. JST *backpropagation* telah dilatihkan dengan variasi algoritma momentum dan menggunakan fungsi aktivasi *log – sigmoid* dan *sum – square error (SSE)* digunakan sebagai indikator kinerja JST. Arsitektur terbaik terdiri dari 3 *input*, 6 *neuron hidden layer*, dan 1 *output*, laju pembelajaran (*learning rate*) 0,1, konstanta momentum yang digunakan adalah 0,2 dan iterasinya sebanyak 20.000.

2.2. Landasan Teori

2.2.1. Demam Berdarah Dengue

1. Pengertian

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 (Arboviroses)* yang sekarang dikenal sebagai *genus Flavivirus, famili Flaviviridae*, yang ditunjukkan pada Gambar 2.1 dan mempunyai 4 jenis serotipe, yaitu ; DEN-1, DEN2, DEN-3, DEN-4.



Gambar 2.1 Virus Dengue Penyebab Penyakit Demam Berdarah
(<http://wikipedia.org>)

DBD adalah penyakit infeksi dengan manifestasi klinis demam, nyeri otot, dan atau nyeri sendi yang disertai penurunan dari sel darah putih, adanya bercak kemerahan di kulit, pembesaran kelenjar getah bening, penurunan jumlah trombosit dan kondisi terberat adalah perdarahan dari hampir seluruh jaringan tubuh. 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 (Sukanto, 2007). Masa inkubasi penyakit ini diperkirakan lebih kurang 7 hari.

Penyakit DBD dapat menyerang semua golongan umur. Sampai saat ini penyakit DBD lebih banyak menyerang anak-anak tetapi dalam dekade terakhir ini terlihat adanya kecenderungan kenaikan proporsi penderita DBD pada orang dewasa.

Indonesia termasuk daerah endemik untuk penyakit DBD. Serangan wabah umumnya muncul sekali dalam 4 - 5 tahun. Faktor lingkungan memainkan peranan bagi terjadinya wabah. Lingkungan dimana terdapat banyak air tergenang dan barang - barang yang memungkinkan air tergenang merupakan tempat ideal bagi penyakit tersebut (Siregar, 2004).

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 yang ditunjukkan pada Gambar 2.2 (Agarkar dan Ghatol, 2010).



Gambar 2.2 Salah satu gejala demam berdarah adalah munculnya ruam pada kulit (<http://wikipedia.org>)

2. Etiologi/Penyebab

Penyebab penyakit DBD adalah virus *Dengue*. Virus ini termasuk kelompok *Arthropoda. Borne Viruses (Arbovirosis)*. Sampai saat ini dikenal ada 4 serotype virus yaitu :

1. Dengue 1 diisolasi oleh Sabin pada tahun 1944.
2. Dengue 2 diisolasi oleh Sabin pada tahun 1944.
3. Dengue 3 diisolasi oleh Sather
4. Dengue 4 diisolasi oleh Sather.

Keempat type virus tersebut telah ditemukan di berbagai daerah di Indonesia dan yang terbanyak adalah type 2 dan type 3. Penelitian di Indonesia menunjukkan Dengue type 3 merupakan serotype virus yang dominan menyebabkan kasus yang berat (Siregar, 2004).

Infeksi oleh salah satu serotype akan menimbulkan antibody terhadap serotype yang bersangkutan, sedangkan antibody yang terbentuk terhadap serotype lain sangat kurang, sehingga tidak dapat memberikan perlindungan yang memadai terhadap serotype lain tersebut (Dinkes Provinsi Jateng, 2004).

3. Penularan

Virus *dengue* ditularkan ke manusia melalui pembawa (*carrier* ataupun *vector*) yaitu gigitan nyamuk *Aedes aegypti* betina yang terinfeksi virus *dengue*. Dan juga dapat ditularkan melalui gigitan nyamuk *Aedes albopictus*. Kedua jenis nyamuk ini ditunjukkan seperti Gambar 2.3. Virus ini bersirkulasi dalam tubuh manusia selama 2 – 7 hari atau selama terjadi demam. Pada masa inilah penularan terjadi.



(a)



(b)

Gambar 2.3 Nyamuk Penyebab Demam Berdarah
(a) Nyamuk *Aedes Aegypti* (b) Nyamuk *Aedes Albopictus*

Apabila penderita digigit oleh nyamuk penular, maka virus *dengue* juga akan terhisap dalam tubuh nyamuk. Virus tersebut kemudian berada dalam lambung nyamuk dan akan memperbanyak diri selanjutnya akan berpindah ke kelenjar ludah nyamuk. Proses tersebut membutuhkan waktu 8 – 10 hari sebelum virus *dengue* dapat ditularkan kembali ke manusia melalui gigitan nyamuk yang terinfeksi. Lama waktu yang dibutuhkan selama masa inkubasi ekstrinsik ini tergantung pada kondisi lingkungan, terutama faktor suhu udara (Hairani, 2009).

4. Diagnosis

Kriteria diagnosa penyakit DBD dapat ditegakkan dengan melihat gejala secara klinis maupun laboratorium. Gejala secara klinis penderita DBD dapat dilihat seperti berikut (Hairani, 2009) :

1. Demam

Demam secara mendadak dan berlangsung selama 2 – 7 hari kemudian kembali kesuhu normal atau bahkan lebih rendah. Demam ini dapat disertai nyeri punggung, nyeri tulang dan sendi, nyeri kepala dan rasa lemah.

2. Pendarahan

Pendarahan biasanya terjadi pada hari kedua hari demam dan pada umumnya terjadi pada kulit dan dapat berupa hasil uji *tourniquet* yang positif.

3. Hapatemogali/Pembesaran Hati

Pada permulaan dari demam biasanya hati sudah teraba, meskipun pada anak yang kurang gizi hati juga dapat teraba. Bila terjadi peningkatan dari hepatomegali dan hati teraba kenyal kemungkinan akan terjadi renjatan pada penderita.

4. Syok/Renjatan

Permulaan renjatan biasanya terjadi pada hari ketiga sejak penderita sakit, dimulai dengan tanda – tanda kegagalan sirkulasi yaitu kulit lembab, dingin pada ujung hidung, jari tangan dan kaki serta sianosis di sekitar mulut. Bila renjatan terjadi pada masa demam maka biasanya menunjukkan prognosis yang buruk. Nadi menjadi lembut dan cepat, kecil bahkan sering tidak teraba. Tekanan darah sistolik akan menurun sampai di bawah angka 80mmHg.

Penegakan penyakit DBD berdasarkan hasil pemeriksaan laboratorium sebagai berikut :

1. *Trombositopeni*

Jumlah trombosit dalam tubuh mengalami penurunan yang drastis hingga mencapai 100.000 sel/mm^3 atau dapat lebih rendah lagi.

2. *Haemoconcentration*

Adanya rembesan plasma karena peningkatan permeabilitas, vaskular, dimanifestasikan dengan hal berikut :

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

5. **Epidemiologi**

Infeksi virus *dengue* dapat menimbulkan penyakit demam yang dikenal juga sebagai demam lima hari. Disebut demikian karena demam yang menyertai penyakit ini terjadi selama lima hari yang disertai dengan nyeri pada sendi, nyeri otot dan nyeri kepala.

Faktor-faktor yang mempengaruhi peningkatan dan penyebaran kasus DBD sangat kompleks, yaitu (1) Pertumbuhan penduduk yang tinggi, (2) Urbanisasi yang tidak terencana & tidak terkendali, (3) Tidak adanya kontrol vektor nyamuk yang efektif di daerah endemis, dan (4) Peningkatan sarana transportasi.

Perkembangan infeksi virus *dengue* dipengaruhi berbagai faktor antara lain status imunitas pejamu, kepadatan vektor nyamuk, transmisi virus *dengue*, keganasan (virulensi) virus dengue, dan kondisi geografis setempat. Dalam kurun waktu 30 tahun sejak ditemukan virus dengue di Surabaya dan Jakarta, baik dalam

jumlah penderita maupun daerah penyebaran penyakit terjadi peningkatan yang pesat. Sampai saat ini DBD telah ditemukan di seluruh propinsi di Indonesia, dan 200 kota telah melaporkan adanya kejadian luar biasa. Incidence rate (IR) meningkat dari 0,005 per 100.000 penduduk pada tahun 1968 menjadi berkisar antara 6-27 per 100.000 penduduk. Pola berjangkit infeksi virus dengue dipengaruhi oleh iklim dan kelembaban udara. Pada suhu yang panas (28-32°C) dengan kelembaban yang tinggi, nyamuk Aedes akan tetap bertahan hidup untuk jangka waktu lama. Di Indonesia, karena suhu udara dan kelembaban tidak sama di setiap tempat, maka pola waktu terjadinya penyakit agak berbeda untuk setiap tempat. Di Jawa pada umumnya infeksi virus dengue terjadi mulai awal Januari, meningkat terus sehingga kasus terbanyak terdapat pada sekitar bulan April-Mei setiap tahun.

6. Klasifikasi

Menurut WHO (1997), derajat parahanya penyakit DBD dibagi menjadi 4 tingkatan, yaitu :

1. Derajat I (Ringan)

Bila demam disertai dengan gejala konstitusional non-spesifik. Satu – satunya manifestasi perdarahan adalah hasil uji torniquet positif dan/atau mudah memar.

2. Derajat II (Sedang)

Bila perdarahan spontan selain manifestasi pasien pada derajat 1, biasanya disertai dengan manifestasi perdarahan kulit, epistaksis, perdarahan gusi, hematemesis atau melena.

3. Derajat III (Berat)

Apabila terjadi kegagalan peredaran darah perifer dimanifestasikan dengan nadi cepat, dan lemah serta penyempitan tekanan nadi atau hipotensi, kulit dingin, lembab, dan gelisah.

4. Derajat IV (Berat sekali)

Bila terjadi renjatan berat dengan tekanan darah tidak terukur, dan nadi tidak terdeteksi.

7. DBD di Indonesia

Penyakit DBD pertama kali ditemukan kasusnya di ibukota Filipina (Manila) pada tahun 1953 yang kemudian menyebar ke berbagai Negara. Di Indonesia penyakit ini mulai ditemukan pada tahun 1968 di Surabaya dan DKI Jakarta yang kemudian menyebar ke berbagai wilayah di Indonesia. Dan kini semua provinsi di Indonesia sudah terjangkit penyakit DBD di berbagai kota maupun desa terutama yang padat penduduknya dan arus transportasinya lancar.

Faktanya selama kurun waktu 1989 s.d 1993, di Indonesia setiap tahun rata – rata 18.000 orang dirawat di Rumah Sakit karena terserang penyakit ini dan 700 – 750 orang diantaranya meninggal dunia (Depkes RI, 1996). Sehingga penyakit DBD di Indonesia merupakan penyakit endemis dari tahun ke tahun angka kejadian dan daerah yang terjangkit terus meningkat serta sering menyebabkan kejadian luar biasa (KLB).

Pada 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%)(Dirjen PP dan PL Depkes RI, 2007).

Berbagai upaya pemerintah Indonesia bersama masyarakat untuk penanggulangan DBD telah dilakukan terutama dengan kegiatan Pemberantasan Sarang Nyamuk (PSN) melalui gerakan 3M (Menguras, Menutup dan Mengubur). Kegiatan PSN telah dilaksanakan secara intensif sejak tahun 2002 yang kemudian gerakan 3M ini ditingkatkan menjadi 3M Plus, dengan cara menggunakan larvasida, memelihara ikan dan mencegah gigitan nyamuk. Berbagai upaya tersebut belum menampakkan hasil yang diinginkan. Salah satu penyebab tidak optimalnya upaya penanggulangan tersebut karena belum adanya perubahan perilaku masyarakat dalam upaya PSN.

2.2.2. Jaringan Syaraf Tiruan (JST)

1. Pengertian

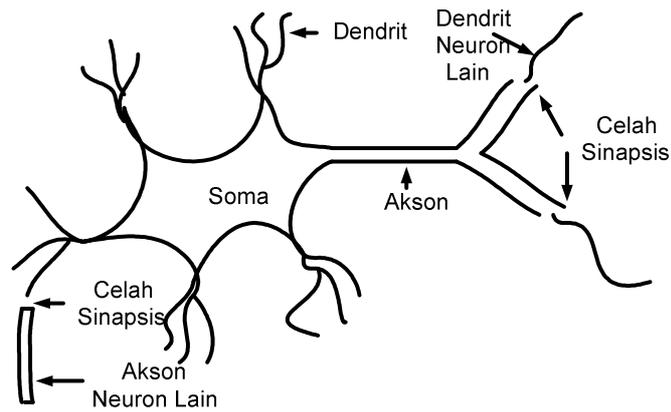
Para ahli pada disiplin ilmu kecerdasan buatan mendefinisikan metode JST ini dengan nama *artificial neural network* (ANN) adalah sebagai model matematika maupun model komputasi yang cara kerjanya merupakan penyederhanaan dari model jaringan biologi otak manusia. Hal ini dikarenakan pada prinsipnya JST merupakan suatu program komputer yang dibuat berdasarkan

cara kerja jaringan biologi otak manusia. Dilihat dari segi fungsi, JST diciptakan untuk merancang suatu komputer yang dapat difungsikan untuk melakukan proses belajar dari suatu contoh kejadian. Sedangkan dari struktur rancangan, JST merupakan suatu rancangan alat penghitung yang ditujukan untuk dapat melakukan serupa dengan apa yang dapat dikerjakan oleh jaringan biologi otak manusia.

JST dapat menangkap pengetahuan dari contoh domain yang dengan mudah baik dari data diskrit maupun kontinu. Serta memiliki kemampuan untuk men-generalisasi yang baik seperti sistem fuzzy. JST adalah model komputasi dari otak manusia. JST berasumsi perhitungan yang didistribusikan melalui beberapa unit sederhana yang disebut *neuron*, yang saling berhubungan dan beroperasi secara paralel sehingga dikenal sebagai sistem pemrosesan paralel terdistribusi atau sistem koneksi (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.

Pengetahuan implisit dibangun dan dilatihkan ke jaringan syaraf. Beberapa JST dapat dilatih oleh pola masukan yang khas dan pola keluaran yang sesuai dengan yang diharapkan. Kesalahan antara *output* dengan aktual diharapkan dapat digunakan untuk memperkuat bobot dari koneksi antara *neuron*. Jenis pelatihan ini dikenal dengan pelatihan terbimbing. Beberapa JST yang lain dilatih dalam model tanpa pengawasan, dimana hanya pola masukan yang diberikan selama pelatihan dan jaringan belajar secara otomatis untuk cluster mereka dalam kelompok dengan fitur yang serupa (Agarkar dan Ghatol, 2010).



Gambar 2.4 Neuron Tunggal

Terdapat tiga komponen penting pada sel syaraf biologis (Gambar 2.4) yang dapat digunakan untuk memahami JST (Siang, 2005), yaitu:

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 mengakumulasikan 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.

JST tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia (*human cognition*) yang didasarkan atas asumsi sebagai berikut (Hermawan, 2006) :

1. Pemrosesan informasi terjadi pada elemen sederhana yang disebut *neuron*.
2. Isyarat mengalir di antara sel syaraf/*neuron* melalui suatu sambungan penghubung.
3. Setiap sambungan penghubung memiliki bobot yang bersesuaian.
4. Setiap sel syaraf akan merupakan fungsi aktivasi terhadap isyarat hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan isyarat keluarannya.

Beberapa sifat JST yang juga dimiliki otak manusia:

1. Pengetahuan yang diperoleh dari proses pelatihan/belajar.

2. Koneksi antar *neuron* dalam otak (sinapsis). Analogi tersebut dalam JST disebut sebagai bobot sinapsis yang berfungsi sebagai penyimpan pengetahuan (memori).
3. Cara belajar otak dianalogikan dengan prosedur algoritma yang memakai algoritma belajar tertentu yang merupakan suatu fungsi untuk mengubah bobot sinapsis jaringan.

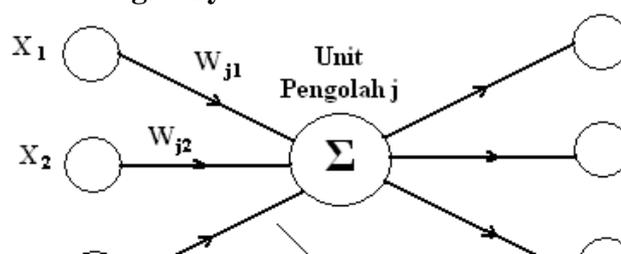
(Hermawan, 2006) Algoritma JST beroperasi secara langsung dengan angka sehingga data yang tidak numerik harus diubah menjadi data numerik. JST tidak dapat menerangkan bagaimana suatu hasil didapatkan. Hal inilah yang membuat JST mampu digunakan untuk menyelesaikan persoalan yang tidak terstruktur dan sulit didefinisikan. JST memiliki banyak kelebihan, antara lain:

1. Kemampuan mengakuisisi pengetahuan walaupun dalam kondisi ada gangguan dan ketidakpastian. Hal ini karena JST mampu melakukan generalisasi, abstraksi, dan ekstraksi terhadap properti statistik dari data.
2. Kemampuan merepresentasikan pengetahuan secara fleksibel. JST dapat menciptakan sendiri representasi melalui pengaturan diri sendiri atau kemampuan belajar.
3. Kemampuan untuk memberikan toleransi atas suatu distorsi (*error/fault*), di mana gangguan kecil pada data dapat dianggap hanya sebagai noise belaka.
4. Kemampuan memproses pengetahuan secara efisien karena memakai sistem parallel, sehingga waktu yang diperlukan untuk mengoperasikannya lebih singkat.

Meskipun demikian JST juga memiliki sejumlah keterbatasan, antara lain kekurangmampuannya dalam melakukan operasi – operasi numerik dengan presisi tinggi, operasi algoritma aritmatik, operasi logika, dan operasi simbolis serta lamanya proses pelatihan yang terkadang membutuhkan waktu berhari – hari untuk jumlah data yang banyak. Karakteristik JST ditentukan oleh:

1. Pola hubungan antar – *neuron* (disebut dengan arsitektur jaringan).
2. Metode penentuan bobot – bobot sambungan (disebut dengan pelatihan atau proses belajar jaringan).
3. Fungsi aktivasi.

2. Pemodelan Jaringan Syaraf Tiruan/Neuron



Gambar 2.5 Struktur Unit Jaringan Syaraf Tiruan (Hermawan, 2006)

Seperti halnya otak manusia, jaringan syaraf juga terdiri dari beberapa *neuron* dan ada hubungan antara *neuron – neuron* tersebut. *Neuron / sel syaraf* adalah sebuah unit pemroses informasi yang merupakan dasar operasi JST. Gambar 2.5 menunjukkan suatu *neuron*.

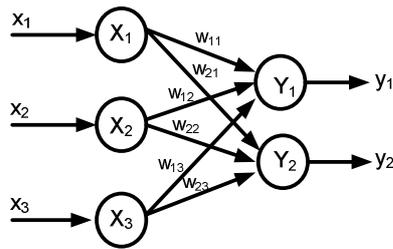
Sebagian besar jaringan syaraf melakukan penyesuaian bobot – bobotnya selama menjalani pelatihan. Pelatihan dapat berupa pelatihan terbimbing (*supervised training*) di mana diperlukan pasangan masukan – sasaran untuk tiap pola yang dilatihkan. Jenis kedua adalah pelatihan tak terbimbing (*unsupervised training*) (Hermawan, 2006). Pada metode ini, penyesuaian bobot (sebagai tanggapan terhadap masukan) tidak perlu disertai sasaran. Sebuah JST mempunyai tiga karakteristik, yaitu:

1. Arsitektur Jaringan

Neuron-neuron dalam suatu sistem jaringan syaraf dikelompokkan dalam lapisan-lapisan. Faktor yang akan menentukan keadaan dari suatu *neuron* adalah fungsi aktivasi dan pola bobotnya. Pada setiap lapisan yang sama, *neuron-neuron* akan memiliki fungsi aktivasi yang sama. Setiap *neuron* pada satu lapisan harus dipetakan tepat ke setiap *neuron* pada lapisan berikutnya (misalkan lapisan *input* ke lapisan *output* pada *single layer*). Adapun arsitektur dari JST antara lain:

a. Jaringan Satu Lapis (*single layer network*)

Pada jaringan satu lapis, hanya memiliki satu lapisan *input* dan lapisan *output*. Pada jaringan ini sinyal masukan langsung diolah menjadi sinyal keluaran, tanpa melalui *hidden layer* seperti contoh yang ditunjukkan pada Gambar 2.6. Pada jaringan satu lapis dapat ditambah dengan bias yang merupakan bobot koneksi dari unit yang aktivasinya selalu 1.



Gambar 2.6 Contoh Jaringan Satu Lapis

Keterangan:

x_1, x_2, x_3 = nilai *input*

X_1, X_2, X_3 = lapisan *input*

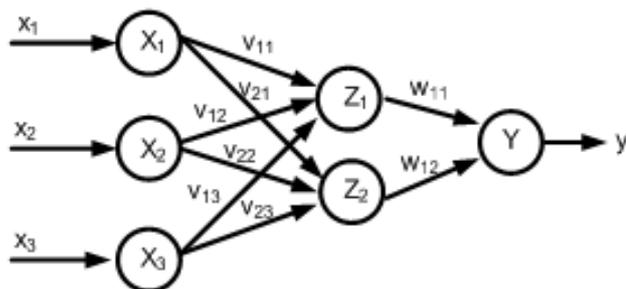
$w_{11}, w_{12}, w_{13}, w_{21}, w_{22}, w_{23}$ = matriks bobot antara *input* dan *output*.

Y_1, Y_2 = lapisan *output*

y_1, y_2 = nilai *output*

b. Jaringan Lapis Banyak (*multi layer net*)

Jaringan lapis banyak terdiri atas satu lapisan *input*, satu lapisan *output* dan satu atau lebih *hidden layer* yang terletak diantara lapisan *input* dan lapisan *output* seperti contoh yang ditunjukkan pada Gambar 2.7. Jaringan dengan banyak lapisan dapat dimanfaatkan untuk menyelesaikan permasalahan yang lebih sulit dan kompleks daripada jaringan dengan satu lapis, akan tetapi tentu saja dengan pembelajaran yang lebih rumit dan lama. Pada jaringan lapis banyak juga dapat ditambah dengan satu buah *neuron* bias pada tiap lapisan.



Gambar 2.7 Contoh Jaringan Lapis Banyak

Keterangan:

x_1, x_2, x_3 = nilai *input*

X_1, X_2, X_3 = lapisan *input*

$v_{11}, v_{12}, v_{13}, v_{21}, v_{22}, v_{23}$ = matriks bobot antara *input* dan *hidden layer*

Z_1, Z_2 = *hidden layer*

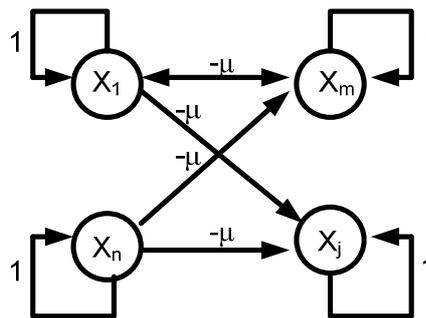
w_1, w_2 = matriks bobot antara *hidden layer* dan *output*

Y = lapisan *output*

y = nilai *output*

c. Jaringan dengan lapisan kompetitif

Hubungan antar *neuron* pada lapisan kompetitif pada umumnya tidak diperlihatkan pada diagram arsitektur. Salah satu contoh dari arsitektur jaringan dengan lapisan kompetitif diperlihatkan pada Gambar 2.8 yang memiliki bobot $-\mu$.



Gambar 2.8 Contoh Jaringan Syaraf dengan Lapisan Kompetitif

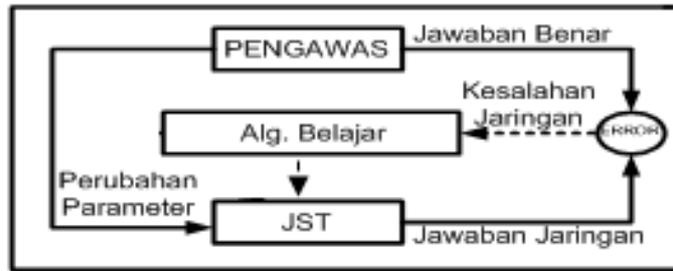
2. Konsep Pembelajaran Jaringan Syaraf Tiruan

Ciri utama dari sebuah sistem JST adalah kemampuan untuk belajar dari contoh ataupun pengalaman terdahulu. Agar sistem dapat berfungsi dengan baik, jaringan tidak diprogram seperti yang dilakukan pada sistem komputer konvensional akan tetapi harus dengan pembelajaran ataupun pelatihan.

Fungsi JST ditentukan oleh bobot-bobot penghubungnya. Hal ini didasarkan pada fungsi masukan keluarannya. Bobot-bobot koneksi pada beberapa kasus dapat berupa variabel yang telah ditentukan pada saat perancangan, tetapi ada juga jaringan yang harus mencari sendiri besarnya bobot-bobot yang sesuai.

Proses penyesuaian parameter secara berurutan dilakukan dengan tujuan untuk mendekati fungsi yang diinginkan. Proses penyesuaian parameter (bobot) inilah dalam JST disebut proses belajar atau pelatihan. Proses belajar dikategorikan dalam dua jenis:

a. Proses belajar Terbimbing (*supervised learning*)



Gambar 2.9 Diagram blok proses belajar dengan pengawasan

Algoritma pembelajaran terbimbing memerlukan keluaran/ target yang telah diketahui sebelumnya, target sebagai dasar pengubahan bobot. Pada proses pelatihan, satu sinyal masukan diberikan kepada satu *neuron* yang selanjutnya akan diolah dan akan menghasilkan satu keluaran. Nilai keluaran ini akan dibandingkan dengan nilai targetnya. Jika terjadi perbedaan antara *output* pembelajaran dengan *output* target, maka disini akan timbul *error*. Untuk mendapatkan nilai keluaran yang sesuai dengan target (*error* yang terjadi kecil), maka harus dilakukan pelatihan secara berkala.

Pada algoritma pembelajaran terbimbing jaringan dibimbing agar dapat menyelesaikan persoalan-persoalan yang terdapat dalam paket belajarnya. Diagram dari konsep belajar terbimbing ditunjukkan pada Gambar 2.9.

b. Proses belajar tidak terbimbing (*unsupervised learning*)

Pada algoritma pembelajaran tak terbimbing, jaringan akan mengubah bobot-bobotnya, sehingga tanggapan terhadap masukan tanpa memerlukan keluaran acuan/ target. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini cocok untuk pengelompokan pola.

3. Fungsi Aktivasi

Fungsi aktivasi merupakan bagian penting dalam tahapan perhitungan keluaran dari suatu algoritma (Hermawan, 2006). Fungsi aktivasi merupakan fungsi matematis yang berfungsi membatasi dan menentukan jangkauan *output* suatu *neuron*. Dengan adanya fungsi aktivasi ini membuat setiap *neuron* yang dipakai pada *neuron* menjadi aktif. Beberapa fungsi aktivasi yang digunakan dalam JST adalah (Hermawan, 2006):

a. Fungsi *Sigmoid*

Fungsi *sigmoid* ada dua, yaitu pertama fungsi *sigmoid* biner, memiliki nilai pada range 0 s.d 1. Fungsi *sigmoid* biner dirumuskan sebagai:

$$f(x) = \frac{1}{1+e^{-\alpha x}} \quad (2.1)$$

dengan turunan dari $f(x)$ adalah:

$$f'(x) = \sigma f(x)[1-f(x)] \quad (2.2)$$

Kedua fungsi *sigmoid* bipolar, memiliki nilai antara -1 s.d 1. Fungsi *sigmoid* bipolar dirumuskan sebagai:

$$f(x) = \frac{1-e^{-\alpha x}}{1+e^{-\alpha x}} \quad (2.3)$$

dengan turunan dari $f(x)$ adalah:

$$f'(x) = \frac{\sigma}{2} [1+f(x)][1-f(x)] \quad (2.4)$$

b. Fungsi *Threshold* (batas ambang)

Fungsi *threshold* ada dua, yaitu:

Fungsi *threshold* biner, yang dirumuskan:

$$F(x) = \begin{cases} 1, & \text{jika } x \geq a \\ 0, & \text{jika } x < a \end{cases} \quad (2.5)$$

Fungsi *threshold* bipolar, yang dirumuskan:

$$F(x) = \begin{cases} 1, & \text{jika } x \geq a \\ -1 & \text{jika } x < a \end{cases} \quad (2.6)$$

dengan a adalah batas ambang (*threshold*). Jika *threshold* a bernilai 0, maka fungsi ini disebut sebagai fungsi undak.

c. Fungsi Identitas (linear)

Fungsi identitas digunakan untuk jaringan yang memiliki keluaran sembarang nilai riil (bukan hanya pada range [0 1] atau [0 -1]). Fungsi identitas memiliki nilai keluaran yang sama dengan nilai masukannya.

$$y = f(x) = x \quad (2.7)$$

d. Fungsi *Saturating Linier*

Fungsi *satrurating linier* dirumuskan sebagai:

$$f(x) = \begin{cases} 1 & \text{jika } x \geq 0,5 \\ x+0,5 & \text{jika } -0,5 < x < 0,5 \\ 0 & \text{jika } x \leq -0,5 \end{cases} \quad (2.8)$$

e. Fungsi Simetrik Saturating Linier

Fungsi simetrik saturating linier dirumuskan sebagai:

$$f(x) = \begin{cases} -1; & \text{jika } x < -1 \\ x; & \text{jika } -1 \leq x \leq 1 \\ 1; & \text{jika } x > 1 \end{cases} \quad (2.9)$$

Jenis fungsi aktivasi yang tepat untuk JST *Backpropagation* adalah sigmoid biner dan sigmoid bipolar.

3. Karakteristik Jaringan Syaraf Tiruan

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).

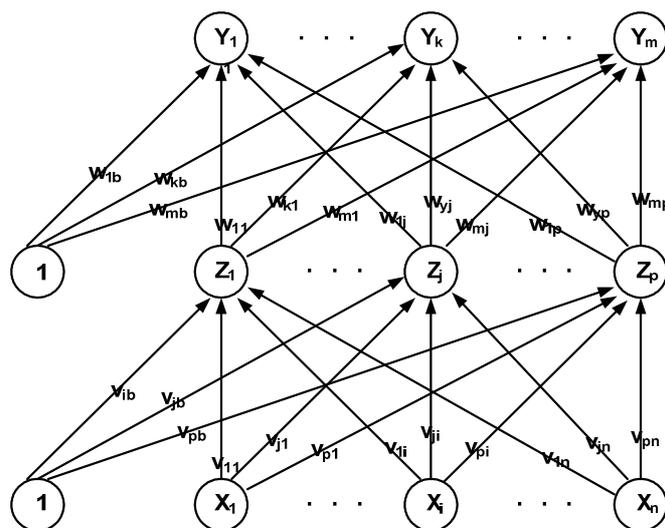
4. Jaringan Syaraf Tiruan Backpropagation

JST *Backpropagation* merupakan salah satu algoritma yang sering digunakan dalam menyelesaikan masalah – masalah yang rumit. Metode ini merupakan metode yang sangat baik dalam menangani masalah pengenalan pola – pola kompleks. Beberapa contoh aplikasi yang melibatkan metode ini adalah kompresi data, deteksi virus komputer, pengidentifikasian objek, sintesis suara dari teks, dan lain – lain. Hal ini dimungkinkan karena jaringan dengan algoritma ini dilatih dengan menggunakan metode belajar terbimbing (Hermawan, 2006).

Prediksi dengan JST melibatkan dua langkah yaitu pelatihan dan langkah yang lain adalah belajar. 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).

1. Arsitektur *Backpropagation*

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih lapis tersembunyi. Gambar 2.10 adalah arsitektur *backpropagation* dengan n buah masukan dengan $1 < i < n$ ditambah sebuah bias, sebuah *hidden layer* yang terdiri dari p unit dengan $1 < j < p$ ditambah sebuah bias, serta m buah unit keluaran dengan $1 < k < m$.



Gambar 2.10 Arsitektur JST Backpropagation

Perambatan maju dimulai dengan memberikan pola masukan ke lapisan masukan. Pola masukan ini merupakan nilai aktivasi unit – unit masukan. Dengan melakukan perambatan maju dihitung nilai aktivasi pada unit – unit di lapisan berikutnya. Pada setiap lapisan, tiap unit pengolah melakukan penjumlahan berbobot dan menerapkan fungsi sigmod untuk menghitung keluarannya (Agarkar dan Ghatol, 2010).

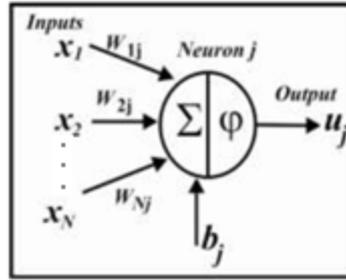
v_{ji} merupakan bobot garis dari unit masukan x_i ke unit lapis tersembunyi z_j (v_{jb} merupakan bobot garis yang menghubungkan bias di unit tersembunyi z_j). w_{kj} merupakan bobot dari unit lapis tersembunyi z_j ke unit keluaran y_k (w_{kb} merupakan bobot dari bias di lapis tersembunyi ke unit keluaran z_k).

2. Algoritma

Dalam algoritma *Backpropagation*, *input* dilewatkan melalui lapisan – lapisan sampai hasil akhir dihitung, dan dibandingkan dengan *output* nyata untuk menemukan kesalahan. Kesalahan ini kemudian disebarkan kembali ke *input* untuk menyesuaikan bobot dan bias dalam setiap lapisan. Dalam rangka untuk mempercepat proses belajar, ada dua parameter dari algoritma *Backpropagation* yang disesuaikan yaitu : laju pembelajaran dan momentum (Agarkar dan Ghatol, 2010).

Laju pembelajaran sangat berpengaruh pada intensitas proses pelatihan. Begitu pula terhadap efektivitas dan kecepatan mencapai konvergensi dari pelatihan. (Hermawan, 2006) nilai laju pembelajaran yang terlalu tinggi dapat menyebabkan jaringan menjadi tidak stabil sedangkan nilai yang terlalu kecil dapat menjadikan waktu belajar yang lama. Oleh karena itu pemilihan konstanta laju pembelajaran harus seoptimal mungkin agar didapatkan proses belajar yang cepat. Nilai laju pembelajaran yang cukup kecil menjamin penurunan gradien terlaksana dengan baik, namun hal ini berakibat bertambahnya jumlah iterasi. Pada umumnya nilai laju pembelajaran dipilih mulai 0,001 sampai dengan 1 selama proses pelatihan. Selain laju pembelajaran, pada metode *Backpropagation* ada koefisien lain yang tujuan penggunaannya untuk mempercepat konvergensi. Nilai konstanta momentum berupa bilangan positif antara 0,5 sampai dengan 0,9 (Purnomo, 2006).

Dalam penelitian ini, digunakan jaringan *Backpropagation* sepenuhnya terhubung dan dilatih. Jaringan terdiri dari lapisan *input* yang mewakili data *input* ke jaringan, *hidden layer* dan lapisan *output* yang mewakili respon dari jaringan. Tiap lapisan terdiri dari sejumlah *neuron*, setiap *neuron* terhubung ke *neuron* lain dari lapisan sebelumnya dan mengadaptasikan bobot w dan bias b seperti yang ditunjukkan dalam gambar 2.11 (Agarkar dan Ghatol, 2010).



Gambar 2.11 Proses Informasi pada JST (Agarkar, 2010)

Untuk menghitung nilai penjumlahan berbobot digunakan rumus persamaan (Hermawan, 2006) :

$$S_j = \sum_{i=0}^n a_i w_{ji} \quad (2.10)$$

dengan :

w_{ji} = bobot sambungan dari unit i ke unit j.

a_i = masukan yang berasal dari unit i.

Setelah nilai S_j dihitung, fungsi sigmoid diterapkan pada S_j untuk membentuk $f(S_j)$. Fungsi sigmoid mempunyai persamaan :

$$f(S_j) = \frac{1}{1 + e^{-S_j}} \quad (2.11)$$

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

Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil.

Langkah 1 : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 – 8.

Langkah 2 : Untuk setiap pasang data pelatihan, lakukan langkah 3-8.

Fase I : Feed Forward

Langkah 3 : Tiap unit *input* menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya.

Langkah 4 : Hitung semua *output* di unit tersembunyi z_j ($j=1, 2, \dots, p$)

$$z_{_netj} = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (2.12)$$

$$z_j = f(z_{_netj}) = \frac{1}{1 + e^{-z_{_netj}}} \quad (2.13)$$

Langkah 5 : Hitung semua *output* jaringan di unit y_k ($k = 1, 2, \dots, m$)

$$y_{\text{net}_k} = w_{ko} + \sum_{j=1}^p z_j w_{kj} \quad (2.14)$$

$$y_k = f(y_{\text{net}_k}) = \frac{1}{1 + e^{-y_{\text{net}_k}}} \quad (2.15)$$

Fase II : Backpropagation

Langkah 6 : Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k ($k = 1, 2, \dots, m$)

$$\delta_k = (t_k - y_k) f'(y_{\text{net}_k}) = (t_k - y_k) y_k (1 - y_k) \quad (2.16)$$

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

$$\Delta w_{kj} = \alpha \cdot \delta_k \cdot z_j \quad (k = 1, 2, \dots, m; j = 0, 1, \dots, p) \quad (2.17)$$

Langkah 7 : Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j ($j = 1, 2, \dots, p$)

$$\delta_{\text{net}_j} = \sum_{k=1}^m \delta_k w_{kj} \quad (2.18)$$

faktor δ unit tersembunyi:

$$\delta_j = \delta_{\text{net}_j} \cdot f'(z_{\text{net}_j}) = \delta_{\text{net}_j} \cdot z_j (1 - z_j) \quad (2.19)$$

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

$$\Delta v_{ji} = \alpha \cdot \delta_j \cdot x_i \quad (j = 1, 2, \dots, p; i = 0, 1, \dots, n) \quad (2.20)$$

Fase III : Perubahan Bobot

Langkah 8 : Hitung semua perubahan bobot Perubahan bobot garis yang menuju ke unit keluaran:

$$w_{kj(\text{baru})} = w_{kj(\text{lama})} + \Delta w_{kj} \quad (k=1,2,\dots,m; j=0,1,\dots,p) \quad (2.21)$$

Perubahan bobot garis menuju ke unit tersembunyi:

$$v_{ji(\text{baru})} = v_{ji(\text{lama})} + \Delta v_{ji} \quad (j=1, 2, \dots, p; i = 0, 1, \dots, n) \quad (2.22)$$

Langkah 9 : Selesai

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

Fase I: Propagasi 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.

Fase II: Propagasi mundur

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.

Fase III: Perubahan bobot

Setelah semua faktor δ dihitung, bobot semua garis dimodifikasikan 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.

3. Optimalitas Arsitektur Backpropagation

Masalah utama yang dihadapi dalam *backpropagation* adalah lamanya iterasi yang harus dilakukan. *Backpropagation* tidak dapat memberikan kepastian tentang berapa *epoch* yang harus dilalui untuk mencapai kondisi yang diinginkan.

a. Pemilihan bobot dan bias awal

Pemilihan inisialisasi bobot akan mempengaruhi apakah jaringan mencapai galat pada minimum global atau lokal. Perubahan bobot antara kedua unit tergantung pada kedua turunan fungsi aktivasi unit di atas dan unit dibawahnya. Sangatlah penting untuk menghindari pemilihan bobot awal yang akan membuat fungsi aktivasi atau turunannya menjadi nol. Jika bobot awal terlalu besar, masukan awal ke tiap unit tersembunyi atau keluaran akan berada pada daerah di mana turunan fungsi *sigmoid* memiliki nilai sangat kecil (disebut daerah jenuh). Sebaliknya jika bobot terlalu kecil sinyal masukan pada unit tersembunyi atau keluaran akan mendekati nol yang menyebabkan pelatihan jaringan akan sangat lambat.

Prosedur umum adalah menginisialisasi bobot dengan nilai acak antara -0,5 dan 0,5 (atau -1 dan 1) atau pada rentang nilai yang lain yang sesuai (Hari Purnomo dan Agus Kurniawan, 2006). Nilai bisa positif atau negatif, karena bobot pada akhir pelatihan juga memiliki kedua tanda tersebut.

b. Jumlah unit tersembunyi

Hasil teoritis yang didapat menunjukkan bahwa jaringan dengan sebuah lapis tersembunyi sudah cukup bagi *backpropagation* untuk mengenali sembarang perkawanan antara masukan dan target dengan tingkat ketelitian yang ditentukan. Akan tetapi penambahan jumlah lapis tersembunyi kadangkala membuat pelatihan lebih mudah.

Semakin banyak *hidden layer* yang digunakan, maka jaringan akan mampu menangani jangkauan statistik yang lebih luas dan tinggi. Jumlah lapisan yang terlalu banyak bisa menyebabkan laju konvergensi menjadi lebih lambat. Hal ini disebabkan sinyal galat berkurang secara numerik jika melewati terlalu banyak lapisan dan lapisan tambahan cenderung menciptakan minimum lokal.

c. Jumlah pola pelatihan

Tidak ada kepastian tentang berapa banyak pola yang diperlukan agar jaringan dapat dilatih dengan sempurna. Jumlah pola yang dibutuhkan dipengaruhi oleh banyaknya bobot dalam jaringan serta tingkat akurasi yang diharapkan.

d. Parameter laju pembelajaran

Parameter laju pembelajaran (*learning rate*) sangat berpengaruh pada proses pelatihan. Begitu pula terhadap efektivitas dan kecepatan mencapai konvergensi dari pelatihan. Nilai optimum dari *learning rate* tergantung permasalahan yang diselesaikan, prinsipnya dipilih sedemikian rupa sehingga tercapai konvergensi yang optimal dalam proses pelatihan.

Nilai *learning rate* yang cukup kecil menjamin penurunan *gradient* terlaksana dengan baik, namun ini berakibat bertambahnya jumlah iterasi. Pada umumnya besarnya nilai laju pembelajaran tersebut dipilih mulai 0,001 sampai 1 selama proses pelatihan.

e. Momentum

Disamping koefisien laju pembelajaran, ada koefisien lain yang bertujuan untuk mempercepat konvergensi dari algoritma *error backpropagation*. Penggunaan koefisien momentum ini disarankan apabila konvergensi berlangsung terlalu lama, dan juga untuk mencegah terjadinya lokal minimum.

Dengan penambahan momentum, bobot baru pada waktu ke- $t+1$ didasarkan atas bobot pada waktu t dan $t-1$. Jika μ adalah konstanta yang menyatakan momentum, maka bobot baru dihitung berdasarkan persamaan (JJ Siang, 2005):

$$w_{kj}(t+1) = w_{kj}(t) + \alpha \cdot \delta_k \cdot z_j + \mu(w_{kj}(t) - w_{kj}(t-1)) \quad (2.23)$$

dan

$$v_{ji}(t+1) = v_{ji}(t) + \alpha \cdot \delta_j \cdot x_i + \mu(v_{ji}(t) - v_{ji}(t-1)) \quad (2.24)$$

5. Sum Square *Error* (SSE) dan Root Mean Square *Error* (RMS)

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:

1. Hitung keluaran jaringan syaraf untuk masukan pertama.
2. Hitung selisih antara nilai keluaran jaringan syaraf dan nilai target / yang diinginkan untuk setiap keluaran.
3. Kuadratkan setiap keluaran kemudian hitung seluruhnya.

Adapun rumusnya adalah:

$$SSE = \sum_p \sum_j (T_{jp} - X_{jp})^2 \quad (2.25)$$

dengan:

T_{jp} = nilai keluaran jaringan syaraf

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

Root Mean Square *Error* (RMS *Error*):

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

Rumus :

$$\text{RMS Error} = \sqrt{\frac{\sum_p \sum_j (T_{jp} - X_{jp})^2}{n_p n_o}} \quad (2.26)$$

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.

2.2.3. Proses Pengembangan Sistem

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 (Whitten, 2004):

1. Mengidentifikasi masalah;
2. Menganalisis dan memahami masalah;
3. Mengidentifikasi persyaratan dan harapan solusi;
4. Mengidentifikasi solusi alternatif dan memilih tindakan terbaik;
5. Mendesain solusi yang dipilih;
6. Mengimplementasi solusi yang dipilih;
7. Evaluasi hasil.

Pengembangan sistem (*systems development*) dapat diartikan menyusun suatu sistem yang baru untuk menggantikan sistem lama secara keseluruhan atau hanya sebagian sistem ataupun memperbaiki sistem yang telah ada. Alasan perlunya memperbaiki sistem antara lain:

1. Adanya permasalahan – permasalahan yang timbul pada sistem lama.
2. Untuk meraih kesempatan – kesempatan.
3. Adanya instruksi dari pimpinan ataupun dari luar organisasi.

1. Siklus Hidup Pengembangan Sistem

Tahapan proses pengembangan sistem, melewati beberapa tahap dari mulai sistem itu direncanakan sampai dengan sistem diterapkan, dioperasikan dan dipelihara. Bila operasi sistem yang sudah dikembangkan masih timbul permasalahan dan tidak dapat diatasi dalam tahap pemeliharaan sistem, maka perlu dikembangkan kembali sistem untuk mengatasinya dan proses ini kembali ke tahap pertama, yaitu tahap perencanaan sistem. Siklus ini disebut dengan siklus hidup suatu sistem (*systems life cycle*).

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*).

2. Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah pengembangan dan penggunaan prinsip pengembangan suara untuk memperoleh perangkat lunak secara ekonomis yang dapat dipercaya dan dapat bekerja secara efisien pada mesin nyata (Pressman, 2001).

Berdasarkan pengertian di atas peneliti menyimpulkan bahwa rekayasa perangkat lunak merupakan suatu cara atau aplikasi dari suatu pendekatan yang sistematis, teratur, terencana, dan dapat diukur terhadap proses pengembangannya, kinerja, dan perawatan perangkat lunak.

3. Paradigma Rekayasa Perangkat Lunak

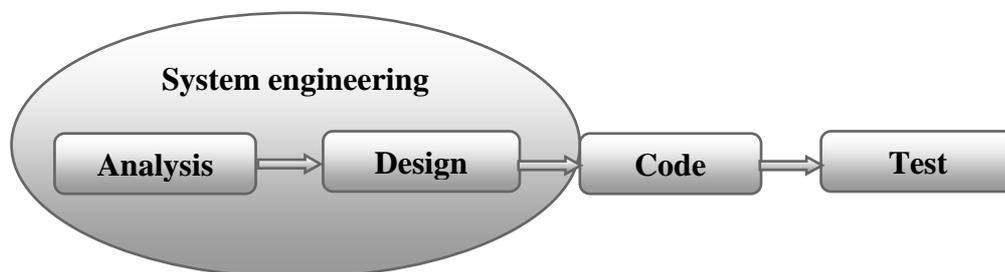
Paradigma rekayasa perangkat lunak atau yang sering dikenal sebagai model proses untuk rekayasa perangkat lunak dipilih berdasarkan sifat aplikasi dan proyeknya, metode, dan alat – alat bantu yang akan dipakai, kontrol, serta penyampaian yang dibutuhkan.

Beberapa contoh model proses rekayasa perangkat lunak:

- a. Model Sekuensial Linier (Linear Sequential Model)
- b. Model prototype (Prototyping Model)
- c. Model RAD (Rapid Application Development Model)
- d. Model Pertambahan (Incremental Model)
- e. Model Spiral (Spiral Model)

4. Model Sekuensial 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). Model sekuensial linier merupakan model proses perangkat lunak yang paling luas dan paling tua. Gambaran dari model proses sekuensial linier ditunjukkan pada Gambar 2.12.



Gambar 2.12 Model Sekuensial Linier (Pressman, 2001)

Tahapan model sekuensial linier meliputi (Pressman, 2001):

- a. Rekayasa dan pemodelan sistem / informasi

Perangkat lunak selalu merupakan bagian dari sistem yang lebih besar, sehingga pekerjaan ini dimulai dengan menetapkan persyaratan untuk semua bagian sistem dan mengalokasikan beberapa sub-set dari persyaratan perangkat lunak ini. Hal ini penting ketika perangkat lunak harus berinteraksi dengan unsure lainnya seperti perangkat keras, orang, dan database.

b. Analisis kebutuhan perangkat lunak

Untuk memahami sifat program yang akan dibangun, seorang analis harus memahami domain informasi untuk perangkat lunak, serta fungsi yang diperlukan, perilaku, kinerja, dan antarmuka yang diperlukan.

c. Desain

Desain perangkat lunak sebenarnya merupakan proses multi langkah yang berfokus pada empat atribut berbeda dari sebuah program yaitu: struktur data, arsitektur perangkat lunak, antarmuka, dan algoritma. Proses desain menerjemahkan kebutuhan menjadi representasi dari perangkat lunak yang dapat dinilai untuk kualitas sebelum coding dimulai.

d. Penerjemahan kode (*code generation*)

Langkah pembuatan kode melakukan tugas menerjemahkan desain ke dalam bentuk yang dapat dibaca oleh mesin.

e. Pengujian (*testing*)

Setelah kode dihasilkan maka proses selanjutnya adalah pengujian. Proses pengujian berfokus pada logika dari perangkat lunak, yaitu melakukan tes untuk mengungkap kesalahan dan memastikan bahwa *input* yang didefinisikan akan menghasilkan hasil yang aktual yang sesuai dengan hasil yang dibutuhkan.

f. Pemeliharaan

Pemeliharaan perangkat lunak menerapkan kembali setiap tahapan program sebelumnya karena terjadi kesalahan tertentu. Misalnya terjadi perubahan akibat sistem operasi yang baru akan tetapi bukan membuat ulang yang baru.

Berdasarkan metode sekuensial linier di atas, beberapa tahapan proses pengembangan sistem meliputi (Gambar 2.13) (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

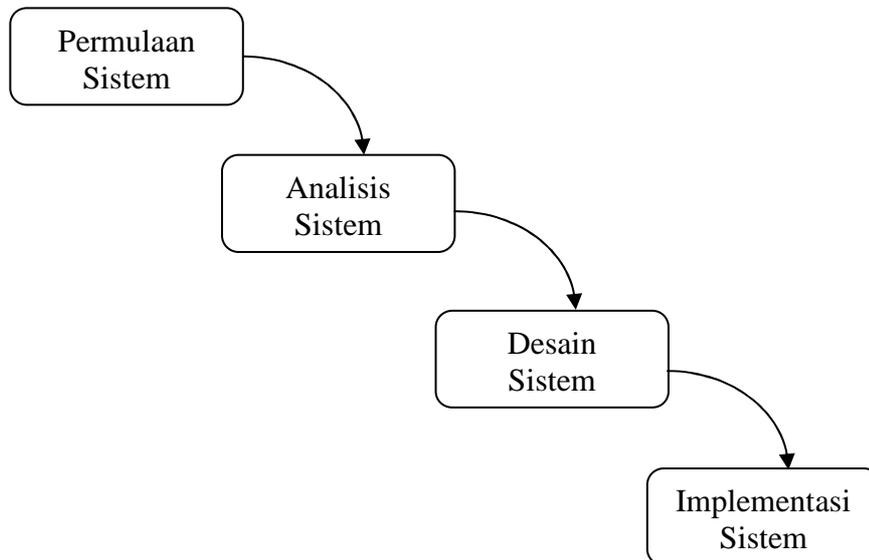
Analisis sistem ini dimaksudkan untuk memberikan pemahaman yang lebih mendalam kepada tim proyek tentang masalah dan kebutuhan yang memicu proyek tersebut. Analisis sistem memerlukan kerja dengan pengguna sistem untuk secara jelas mendefinisikan kebutuhan bisnis dan harapan untuk setiap sistem baru yang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

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.



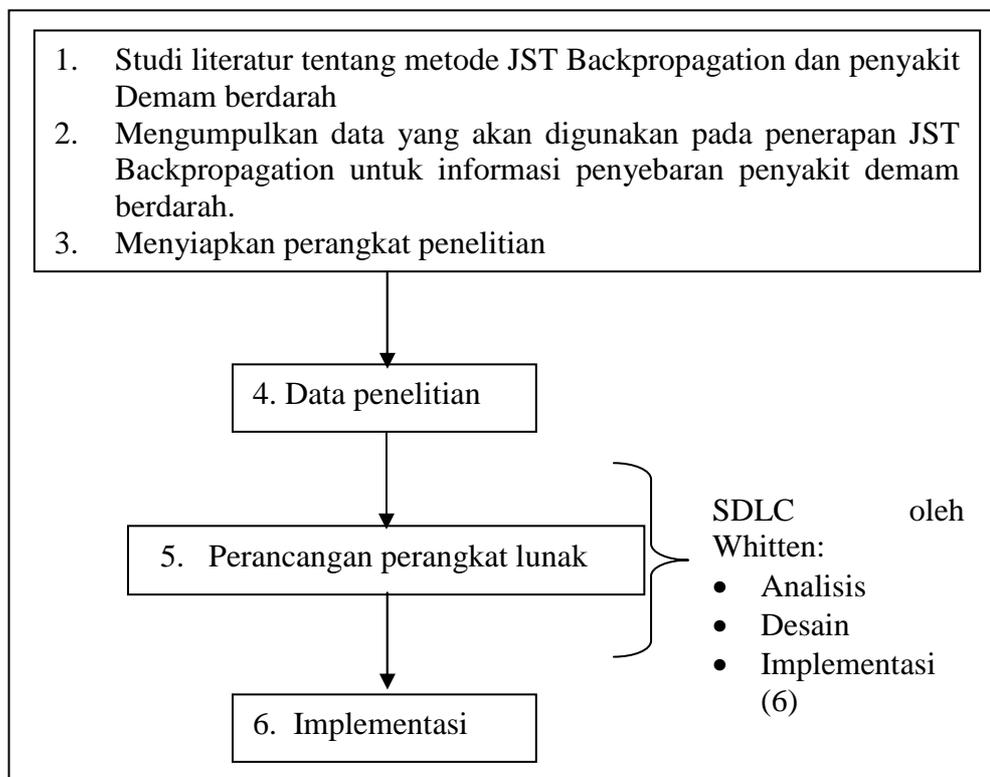
Gambar 2.13 Metode Sekuensial atau Waterfall (Whitten, 2004)

BAB III METODE PENELITIAN

3.1. Desain Penelitian

Desain penelitian merupakan tahapan yang merepresentasikan langkah – langkah yang akan dilakukan dalam melaksanakan penelitian, tujuannya untuk memudahkan peneliti dalam melakukan penelitian.

Berikut adalah tahapan penelitian yang akan dilakukan peneliti dalam proses penelitian yang berjudul “Sistem Informasi Penyebaran Penyakit Demam Berdarah menggunakan Metode Jaringan Syaraf Tiruan Backpropagation” dapat dilihat secara jelas pada Gambar 3.1 yang menunjukkan rencana atau struktur penelitian yang akan digunakan untuk memecahkan permasalahan dalam penelitian ini.



Gambar 3.1 Desain Penelitian

Tahapan – tahapan desain penelitian yang peneliti lakukan diantaranya adalah:

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

3.2. Metode Penelitian

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

3.2.1. Metode Pengumpulan Data

1. Metode Studi Literatur

Yaitu metode dengan mengumpulkan dan mempelajari literatur yang berkaitan dengan metode Jaringan Syaraf Tiruan Backpropagation untuk informasi, dan pembahasan mengenai penyebaran penyakit demam berdarah. Metode ini bersumber dari jurnal – jurnal ilmiah, paper, artikel, buku, serta sumber ilmiah lainnya.

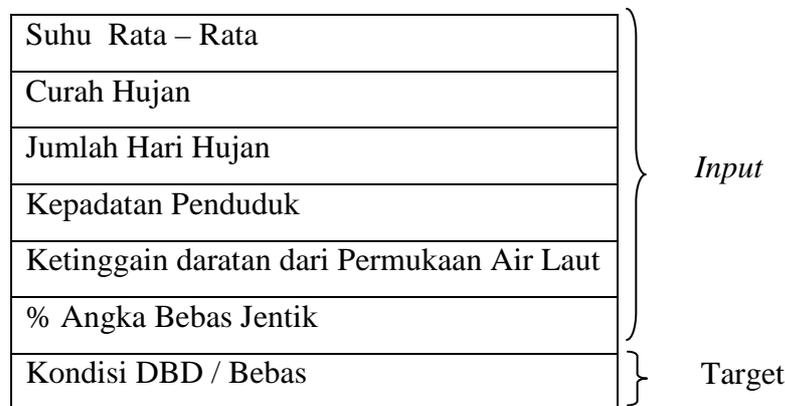
2. Metode Wawancara

Guna memperoleh data seakurat mungkin, proses wawancara dengan pakar ataupun ahli perlu dilakukan secara langsung dengan pihak terkait yang menangani langsung dengan objek yang diteliti, dalam hal ini adalah penyebaran penyakit demam berdarah pada wilayah Kabupaten Brebes. Objek wawancaranya adalah seorang ahli epidemiologi yang menangani langsung penyakit demam berdarah di Kabupaten Brebes yaitu Bapak Awaludin SKM, M.Kes. Sehingga dari wawancara tersebut diperoleh data dan informasi yang dapat membantu proses penelitian ini.

Langkah – langkah pengumpulan data yang dilakukan pada penelitian kali ini adalah meliputi :

- a. Membuat rancangan data input dan data output yang akan digunakan 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 (%).



Gambar 3.2 Pasangan *Input* dan Target Sistem

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 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 3.2.

Informasi penyebaran penyakit DBD dengan menggunakan jaringan syaraf tiruan akan semakin baik apabila data historis yang digunakan dalam proses pelatihan semakin banyak. Hal ini akan meningkatkan kemampuan sistem dalam memberikan informasi suatu wilayah terhadap penyebaran penyakit DBD. 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 (*Lampiran 1*).

b. Membagi data ke dalam dua kelompok data yaitu data set pelatihan dan data set pengujian

Data yang berfungsi sebagai masukan dan keluaran yang diperoleh dari penelitian dibagi menjadi dua kelompok besar, yaitu data 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 x 85 untuk data pelatihan dan 17 set data digunakan sebagai data pengujian dengan ukuran matriks 7 x 17.

Tabel pada *Lampiran 2* merupakan data faktor lingkungan yang mempengaruhi penyebaran penyakit DBD di 17 kecamatan yang ada di kabupaten Brebes sebagai data pelatihan dan informasi penyebaran penyakit DBD sebagai target ataupun keluaran sebelum diolah (data primer pelatihan). Data tersebut berjumlah 85 set data dan bersumber dari pemerintah Kabupaten Brebes dari tahun 2005 – 2009.

Tabel *Lampiran 3* merupakan data faktor lingkungan yang mempengaruhi penyebaran penyakit DBD di 17 kecamatan yang ada di kabupaten Brebes sebagai data pengujian dan informasi penyebaran penyakit DBD sebagai target ataupun keluaran sebelum diolah (data primer pengujian). Data tersebut berjumlah 17 set data dan bersumber dari pemerintah Kabupaten Brebes tahun 2010.

Tabel pada *Lampiran 4* merupakan data faktor lingkungan yang mempengaruhi penyebaran penyakit DBD di 17 kecamatan yang ada di kabupaten Brebes sebagai data pelatihan dan informasi penyebaran penyakit DBD sebagai target ataupun keluaran setelah diolah (normalisasi) ke dalam bentuk numerik antara 0 s.d. 1. Sedangkan untuk data yang akan digunakan sebagai data pengujian sistem informasi penyebaran penyakit DBD di 17 kecamatan yang ada di kabupaten Brebes yang telah diolah (normalisasi) ataupun telah diubah ke dalam bentuk numerik yang berkisar antara 0 s.d. 1 terdapat pada *Lampiran 5*.

3.2.2. Metode Pengembangan Sistem

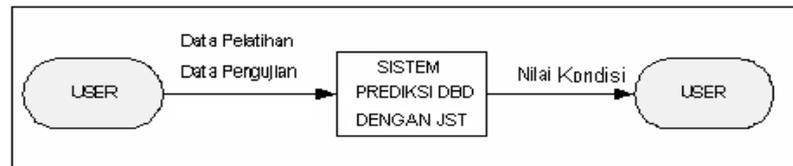
Pengembangan sistem meliputi proses – proses terstruktur antara lain analisis, desain, implementasi, pemeliharaan dan pengujian yang dituangkan dalam suatu metode dengan nama *System Development Life Cycle (SDLC)*.

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

1. Rekayasa dan Pemodelan Sistem

Untuk menggambarkan sistem informasi penyebaran penyakit DBD dengan metode JST *Backpropagation* yang dikembangkan, perlu dibuat terlebih dahulu diagram alir sistem. Adapun diagram alir sistem yang akan digunakan, meliputi:

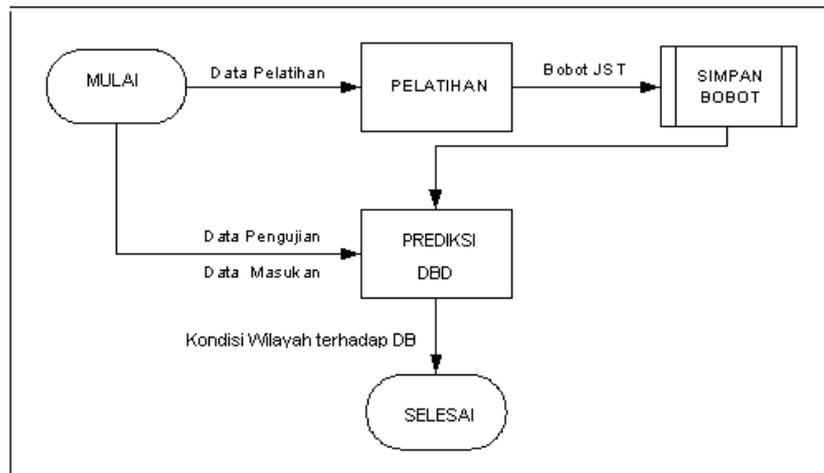
- Diagram Alir Sistem Level 1



Gambar 3.3 Diagram alir level 1

Pada Gambar 3.3 dapat dijelaskan bahwa interaksi *user* terhadap sistem dapat dilihat dari masukan yang diberikan *user* ke dalam sistem dan keluaran sistem yang akan dimanfaatkan oleh *user*. Data-data yang dimasukkan oleh *user* kepada sistem adalah:

1. Data pelatihan, terdiri atas 85 pasang pola dari total data yang tersedia yang digunakan untuk proses pelatihan. Data ini bersumber dari data tahunan dari tahun 2005 – 2009.
 2. Data pengujian, terdiri atas 17 pasang pola dari data yang tersedia yang tidak diikuti dalam proses pelatihan. Data ini bersumber dari data tahun 2010. Fungsi dari data pengujian adalah untuk mengetahui apakah sistem sudah mampu mengenali pola data pelatihan yang diberikan. Pada data pelatihan dan data pengujian target sudah diketahui.
 3. Data masukan, merupakan data faktor-faktor yang berpengaruh terhadap kondisi seseorang yang dimasukkan oleh *user* untuk keperluan diagnosis kondisi wilayah terhadap penyakit DBD. Pada data masukan belum diketahui targetnya.
- Diagram Alir Sistem Level 2



Gambar 3.4 Diagram alir level 2

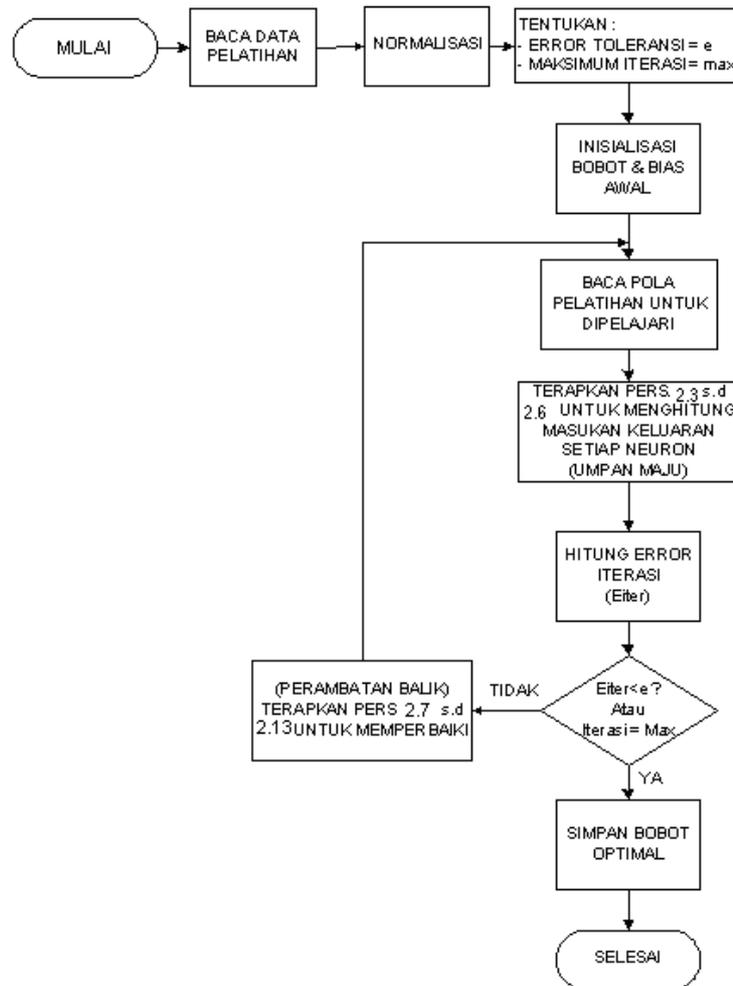
Pada diagram alir level 2 (Gambar 3.4) menjelaskan mengenai proses utama yang ada pada sistem. Saat *user* mulai memberikan masukan data pelatihan, maka proses pelatihan dikerjakan untuk mendapatkan bobot yang optimal. Setelah bobot optimal diperoleh, kemudian bobot ini disimpan untuk proses pengujian dan informasi kondisi wilayah terhadap kasus penyakit DBD.

User memasukkan data pengujian untuk melakukan pengujian yang akan menghasilkan hasil pengujian berupa nilai kondisi wilayah. Hasil pengujian akan dibandingkan dengan target yang ada untuk mengetahui kemampuan sistem dalam mempelajari data latih yang diberikan kepadanya.

- Diagram Alir Sistem Level 3

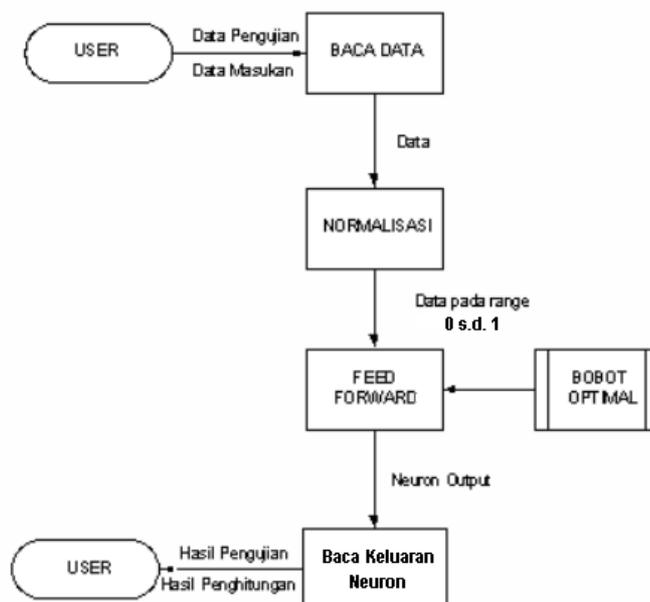
Diagram alir level 3 terbagi menjadi dua yaitu:

- Diagram alir level 3 proses pelatihan (Gambar 3.5).



Gambar 3.5 Diagram alir proses pelatihan

- o Diagram alir level 3 proses pengujian dan informasi kondisi wilayah terhadap penyakit DBD ataupun normal (Gambar 3.6).



Gambar 3.6 Diagram alir proses pengujian

2. Analisis

Dalam tahapan analisis, hal – hal yang harus diperhatikan antara lain adalah ukuran jaringan yang dirancang, *input* dan *output*, tipe pelatihan, dan waktu operasi rutin dari sistem berjalan. Mengenai kemampuan JST dapat dilihat pada Tabel 3.1 (Puspitaningrum, 2006).

Tabel 3.1 Kemampuan JST

Paradigma	Metode Pelatihan	Waktu Pelatihan	Waktu Eksekusi Pengujian
Backpropagation	Terbimbing	Lambat	Cepat
ART2	Tidak terbimbing	Cepat	Cepat
Kohonen	Tidak terbimbing	Sedang	Cepat
Hopfield	Terbimbing	Cepat	Sedang
Boltzman	Terbimbing	Lambat	Lambat

Untuk dapat menginformasikan 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 menginformasikan 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.

3. Desain

A. Desain JST

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

1. Tingkat Neuron

Pada sistem informasi penyebaran penyakit DBD ini menggunakan variabel *input* berdasarkan data yang diperoleh dari penelitian. Data yang diperoleh merupakan data yang berupa bilangan real. Data tersebut kemudian dinormalisasi. Sedangkan untuk *outputnya* adalah nilai kondisi dari suatu wilayah terhadap kasus DBD yang direpresentasikan dengan bilangan biner 0 untuk nilai kondisi bebas dari kasus DBD dan 1 untuk nilai kondisi terdapat kasus DBD.

Fungsi aktivasi yang akan digunakan adalah fungsi aktivasi sigmoid biner dengan alasan bahwa fungsi aktivasi ini merupakan fungsi aktivasi yang paling umum digunakan untuk metode JST *Backpropagation*.

2. *Tingkat Jaringan*

Pada tingkat jaringan dilakukan penentuan banyaknya lapisan. Seperti telah dinyatakan sebelumnya bahwa JST *backpropagation* merupakan jaringan multilayer dalam pengertian bahwa jaringan dapat memiliki *hidden layer* yang berjumlah lebih dari satu. Dalam penelitian ini untuk menentukan jumlah *hidden layer* maupun jumlah *neuron* tiap *hidden layer* dilakukan secara konstruktif yaitu dari satu *hidden layer* dengan jumlah *neuron* tiap *hidden layer* yang divariasi. Secara umum JST tersusun atas 3 lapisan meliputi : lapisan *input*, lapisan *hidden layer*, dan lapisan *output*.

JST tersebut direncanakan hanya akan menerima *input* yang berkisar antara 0 sampai dengan 1 sehingga data penelitian harus diatur sebagai sekumpulan angka. Hal ini bertujuan untuk mempermudah proses komputasi.

3. *Tingkat Pelatihan*

Rancangan jumlah *neuron* pada JST *Backpropagation* untuk sistem informasi 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 informasi 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 seperti yang ditunjukkan pada Tabel 3.2.

Tabel 3.2 Representasi nilai *output*

<i>Output</i>	Representasi
0	Bebas

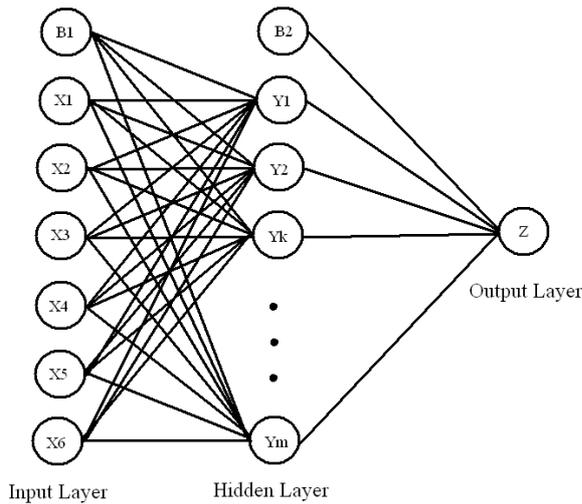
1	Ada kasus DBD
---	---------------

Berdasarkan kemampuan JST yang ditunjukkan pada Tabel 3.1 dapat dilihat bahwa JST *Backpropagation* memiliki pelatihan yang lambat. Tujuan dari penerapan JST *Backpropagation* adalah untuk mendapatkan kesetimbangan antara respons yang benar terhadap pola pelatihan (kemampuan memorisasi) dan respons yang baik dan benar terhadap pola pelatihan baru (kemampuan menggeneralisasi) maka tidak begitu penting untuk melanjutkan proses pelatihan sampai nilai *error*nya mencapai nilai minimum. Oleh sebab itu nilai *error* yang dapat diterima jaringan atau yang sering disebut dengan galat *error* harus ditentukan terlebih dahulu dengan tujuan untuk menghentikan pelatihan apabila *error* jaringan lebih kecil atau sama dengan nilai galat *error* yang telah ditentukan tersebut. Pada aplikasi ini nilai batas *error* yang direncanakan akan digunakan adalah sebesar 0,001. Nilai batas *error* dipilih yang bernilai kecil untuk tetap menjaga keakuratan hasil yang diperoleh.

B. Perancangan JST Backpropagation untuk DBD

Pada penelitian ini menggunakan bahasa pemrograman Matlab. Berikut diuraikan langkah – langkah pengembangan simulasi sistem informasi penyebaran penyakit DBD menggunakan Matlab.

1. Arsitektur Jaringan



Gambar 3.7 Arsitektur JST untuk Sistem Informasi Penyebaran Penyakit DBD

Pengembangan sistem informasi penyebaran penyakit DBD dengan metode JST *Backpropagation* dimulai dari perancangan arsitektur jaringan seperti terlihat pada Gambar 3.7.

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.

Tiap *neuron* pada lapisan masukan menerima sinyal masukan berupa nilai faktor – faktor lingkungan yang berpengaruh terhadap penyebaran penyakit DBD dan meneruskannya ke *neuron* pada *hidden layer* di atasnya. Jumlah *neuron* pada lapisan masukan sesuai dengan jumlah variabel yang dipilih sebagai masukan jaringan dan ditambah satu buah *neuron* bias. Pada Gambar 3.7 terlihat bahwa jumlah *neuron* lapisan masukan antara 1 sampai dengan 6 sesuai dengan jumlah faktor lingkungan yang akan dijadikan data *inputan* pada sistem ditambah satu buah *neuron* bias.

Untuk menentukan jumlah *neuron* pada *hidden layer* tersebut dilakukan penelitian terhadap jaringan dengan satu *hidden layer* dengan jumlah *neuron* untuk *hidden layer* sebanyak m *neuron* ditambah satu *neuron* bias dengan fungsi aktivasi sigmoid biner. Lapisan keluaran terdiri dari sebuah *neuron* yang menghasilkan keluaran berupa kondisi wilayah terhadap penyakit DBD, dengan fungsi aktivasi yang digunakan adalah fungsi sigmoid biner.

Bobot dan bias awal diinisialisasi dengan bilangan acak kecil antara -1 hingga 1. Bobot awal akan mempengaruhi jaringan akan mencapai titik minimum lokal atau minimum global, dan lama konvergensinya. Bobot awal yang terlalu besar membuat nilai turunan fungsi aktivasi menjadi sangat kecil, hal ini menyebabkan perubahan bobotnya menjadi sangat kecil juga (Siang, 2004). Ukuran matriks bobot lapisan *input* adalah $6 \times m$, dan untuk *hidden layer* pertama adalah $m \times 1$.

Hal lain yang perlu diperhatikan juga adalah parameter yang harus diatur dalam jaringan meliputi:

1. Laju pembelajaran

Laju pembelajaran dipilih mulai dari 0,001 hingga 1 selama pelatihan (Hari Purnomo, 2006).

2. Momentum

Konstanta momentum berupa bilangan positif antara 0,5 hingga 0,9 (Hari Purnomo, 2006).

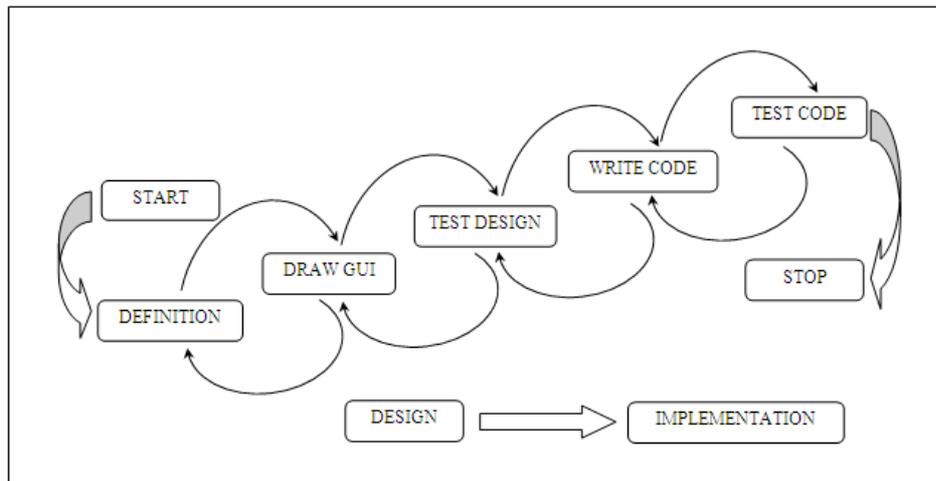
3. Kinerja tujuan

Kinerja tujuan adalah target nilai fungsi kinerja, iterasi akan dihentikan apabila nilai fungsi kinerja kurang dari atau sama dengan kinerja tujuan.

4. Jumlah iterasi maksimum

Maksimum iterasi adalah jumlah *epoch* maksimum yang boleh dilakukan selama proses pelatihan. Iterasi akan dihentikan apabila jumlah *epoch* yang sudah dilatihkan melebihi jumlah maksimum *epoch*.

2. Desain Tampilan berbasis GUI Matlab

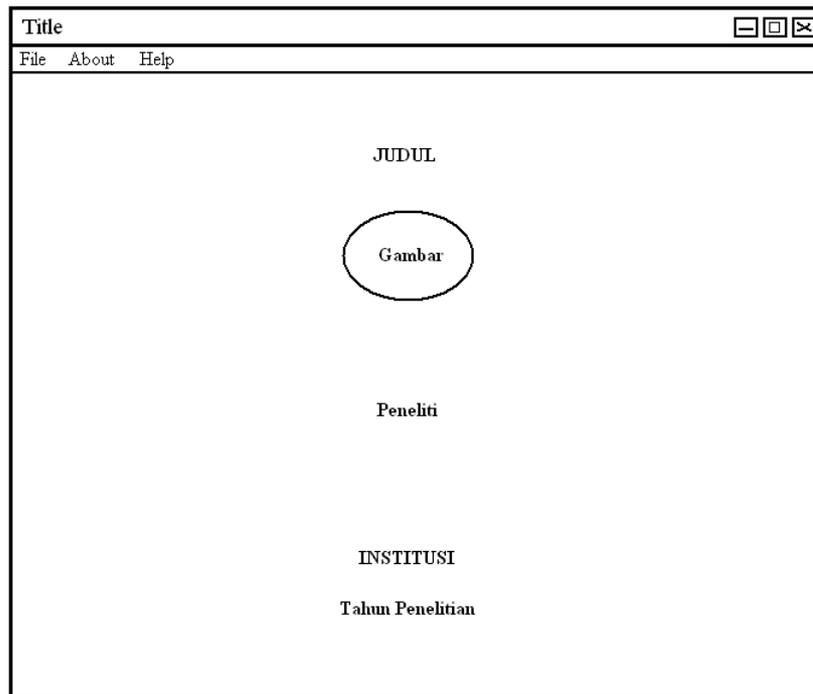


Gambar 3.8 Proses Perancangan GUI

Graphic User Interface (GUI) pada dasarnya merupakan media tampilan grafis sebagai pengganti perintah teks untuk *user* berinteraksi (Gunaidi, 2006). Dalam sistem informasi penyebaran penyakit DBD yang dikembangkan ini meliputi beberapa tampilan, yaitu tampilan utama, tampilan pelatihan dan pengujian data, serta tampilan informasi. Pada proses pembuatan GUI dapat dituangkan ke dalam tahap perancangan (design) ke tahap implementasi (Hermawan, 2006). Konsep ini diilustrasikan seperti Gambar 3.8.

Tampilan Utama

Pada tampilan utama berisi judul program dan beberapa menu pilihan yaitu pertama “Menu File”, yang terdiri dari “Menu Sistem” berfungsi untuk menuju ke tampilan proses pelatihan dan pengujian data, “Menu Print”, untuk mencetak tampilan utama dan “Menu Exit” untuk keluar dari program. Kedua “Menu Help”, berfungsi untuk menuju ke tampilan help yang berisi petunjuk penggunaan sistem. Rancangan/kerangka tampilan utama terlihat pada Gambar 3.9.



Gambar 3.9 Kerangka Tampilan Menu Utama

Tampilan Pelatihan dan Pengujian Data

Menu pelatihan, dan pengujian berfungsi untuk melakukan proses pelatihan jaringan, selanjutnya adalah proses pengujian jaringan dan yang terakhir adalah proses prediksi. Pada menu ini ada empat bagian utama, yaitu pertama arsitektur jaringan yang berisi parameter yang harus diisi oleh user yaitu *input* sistem, jumlah *neuron* pada *hidden layer* pertama. Kedua parameter jaringan, yang berisi parameter jaringan yang harus diisi oleh user yaitu konstanta laju pembelajaran, konstanta momentum, target *error*, maksimum iterasi, serta periode iterasi. Ketiga analisis *output* jaringan, yang terdiri dari nilai MSE, m, b, dan r. Keempat adalah layar tampilan *output* jaringan secara grafis. Rancangan tampilan menu pelatihan dan pengujian seperti pada Gambar 3.10.

Gambar 3.10 Kerangka Tampilan Menu Pelatihan dan Pengujian

Tampilan Informasi

Pada menu ini ada dua bagian utama, yaitu pertama bagian faktor lingkungan yang mempengaruhi yang berisi parameter yang harus diisi oleh user yaitu suhu rata – rata, curah hujan, jumlah hari hujan, ketinggian wilayah, kepadatan penduduk, serta %ABJ. Nilai yang dimasukkan adalah nilai sembarang. Kedua adalah tampilan untuk informasi tingkat endemis suatu wilayah berdasarkan intensitas jumlah kasus dalam 3 tahun terakhir. Rancangan tampilan menu informasi terlihat pada Gambar 3.11.

JUDUL

Faktor yang mempengaruhi :

Suhu rata - rata :

Curah hujan :

Jumlah hari hujan :

Ketinggian wilayah :

Kepadatan Penduduk :

%ABJ :

Prediksi

Hasil Prediksi

Prediksi Tingkat Endemis :

Tahun 1 :

Tahun 2 :

Tahun 3 :

Tingkat Endemisitas

Hasil Prediksi

Reset

Keluar

Gambar 3.11 Kerangka Tampilan Menu Informasi

4. Implementasi

Tujuan dari tahap ini adalah untuk menghasilkan sistem yang dapat diterapkan untuk menginformasikan penyebaran penyakit DBD yang akurat dan konsisten. Tahap implementasi ini meliputi beberapa tahap, meliputi:

a. *Input Data*

Preview of F:\Utiah S2 MSI Undip\Semester IV\THESIS\DENGUE RESEARCH\FINAL PROJECT THESIS\LAH DATA\SOURCE CODING THESIS\cooling final 2 april 2012\p2.dat						
1	2	3	4	5	6	7
1	24.0	113.0	2371.0	367.0	500.0	75.0
2	26.2	149.0	2566.0	447.0	161.0	100.0
3	26.2	211.0	3094.0	1377.0	162.0	54.0
4	25.2	156.0	4747.0	876.0	342.0	100.0
5	22.2	113.0	2371.0	906.0	875.0	97.0
6	26.2	126.0	2576.0	844.0	175.0	100.0
7	26.7	100.0	2141.0	833.0	23.0	68.0
8	26.7	99.0	2101.0	878.0	17.0	83.0
9	26.7	127.0	2447.0	825.0	22.0	97.0
10	26.8	99.0	1745.0	1374.0	5.0	78.0
11	26.8	99.0	2234.0	1345.0	3.0	69.0
12	26.8	113.0	2371.0	2341.0	11.0	79.0
13	26.8	99.0	1689.0	1549.0	3.0	92.0
14	26.9	114.0	2234.0	1666.0	1.0	74.0
15	26.8	124.0	2102.0	1446.0	5.0	45.0
16	26.8	36.0	756.0	2382.0	5.0	46.0
17	26.8	124.0	1633.0	1884.0	3.0	39.0
18	24.2	118.0	2467.0	369.0	500.0	74.0
19	26.4	133.0	2821.0	447.0	161.0	100.0
20	26.4	157.0	1688.0	1387.0	162.0	56.0
21	25.4	140.0	4034.0	877.0	342.0	100.0
22	22.4	118.0	2467.0	904.0	875.0	100.0
23	26.4	102.0	2523.0	846.0	175.0	100.0

Gambar 3.12 Matriks Data Pelatihan disimpan dengan nama p2.dat

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* seperti yang ditunjukkan Gambar 3.12.

b. Normalisasi

Sebelum digunakan untuk proses pelatihan, perlu dilakukan penskalaan terhadap harga-harga masukan dan target sedemikian hingga data-data masukan dan target tersebut masuk 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):

$$A_i' = \frac{A_i}{A_{\max}} \quad (3.1)$$

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. Untuk keperluan ini kita tinggal menuliskan persamaan (3.1) ke dalam script Matlab editor seperti berikut:

```
-----  
%Preprocessing  
Anormal=P/Amaks3  
-----
```

Dengan P adalah matriks *input* dan $Amaks3$ adalah nilai tertinggi dari sekumpulan data *input* P . Fungsi di atas akan menghasilkan $Anormal$ yang merupakan matriks *input* yang ternormalisasi. Tujuan utama normalisasi adalah agar terjadi sinkronisasi data, disamping itu juga untuk memudahkan dalam proses komputasi. Hasil dari normalisasi adalah sekumpulan bilangan yang berkisar antara 0 dan 1. Dengan menggunakan persamaan (3.1) maka matrik data pelatihan yang telah dinormalisasi ditunjukkan seperti pada **Lampiran 6**.

c. Komputasi Backpropagation

Pada bagian inilah proses dari JST backpropagation (Gambar 3.3). Semua rumus dan algoritma dari backpropagation yang telah dijelaskan pada (2.2.2.4). Sedangkan nilai awal dari penimbang / pembobot adalah berkisar antara -0,5 sampai 0,5. Nilai pembobot awal tidak boleh terlalu besar karena nilai turunan fungsi aktivasinya menjadi sangat kecil juga. Hal ini perlu dihindari karena akan menyebabkan perubahan bobotnya menjadi sangat kecil. Akan tetapi pada fase III perubahan bobot berdasarkan momentum dengan rumusan yang tercantum dalam persamaan (2.23) dan (2.24).

Untuk mencapai tujuan ini perlu dipahami proses – proses yang terjadi di dalam JST *Backpropagation*. Proses tersebut meliputi:

1. Inisialisasi Bobot dan bias

Seluruh bobot dan bias diset ke dalam bilangan acak yang kecil.

- a. Bobot – bobot antara lapisan *input* dengan *hidden layer* disebut sebagai V_{ik} sejumlah $6 \times m$ dengan $i=1, \dots, n$ dan $m=1, \dots, n$
- b. Bobot – bobot bias antara lapisan *input* dengan *hidden layer* disebut V_{om} sejumlah $1 \times m$ dengan m adalah jumlah *neuron* pada *hidden layer*.
- c. Bobot – bobot antara *hidden layer* dengan lapisan *output* disebut W_{om} sejumlah $Y_m \times Z$ dengan o adalah jumlah *neuron* lapisan *output* =1 dan m adalah jumlah *neuron hidden layer*.
- d. Bobot – bobot bias antara *hidden layer* pertama dengan lapisan *output* disebut W_{ob} sejumlah 1 buah

Untuk menghindari minimum lokal, keseluruhan bobot di atas diinisialisasi dengan menggunakan bilangan acak kecil. Matlab menyediakan fungsi untuk inisialisasi bobot dan bias seperti di bawah ini:

```
-----  
%Initialization Weight and biases  
BobotAwal_Input=net.IW{1,1}  
BobotAwal_bias_Input=net.b{1,1}  
BobotAwal_Lapisan1=net.LW{2,1}  
BobotAwal_bias_Lapisan1=net.b{2,1}  
-----
```

`Net.IW{1,1}` untuk menentukan bobot awal random untuk lapisan pertama, sedangkan `net.b{1,1}` untuk menentukan nilai random bias. Kemudian untuk `net.IW{2,1}` untuk menentukan nilai random untuk bobot pada lapisan kedua (*hidden layer*) dari lapisan pertama (*input*). Untuk nilai bias juga sama dengan algoritma penentuan nilai bobot. Dengan menggunakan data pelatihan *p2.dat* maka sistem akan menginisialisasi nilai bobot dan bias. Proses pelatihannya meliputi:

- Membangun JST *Backpropagation*, Matlab menyediakan fungsi sebagai berikut:

```
net = newff(minmax(Anormal),[S1 S2],{TF1 TF2},BTF)
```

`minmax(Anormal)`: matriks berukuran $R \times 2$ yang berisi nilai minimum dan maksimum variabel *input*, dengan *Anormal* adalah matriks *input* yang sudah dinormalisasi.

`S1` = jumlah *neuron* pada *hidden layer* pertama.

`S2` = jumlah *neuron* pada lapisan *output*.

TF1 = fungsi aktivasi pada hidden layer pertama

TF2 = fungsi aktivasi pada hidden layer kedua

BTF = fungsi pelatihan jaringan .

- Inisialisasi parameter pelatihan yang meliputi laju pembelajaran, momentum, goal, serta jumlah iterasi. Untuk menginisialisasi parameter – parameter tersebut Matlab menyediakan fungsi sebagai berikut:

```
-----  
%Set parameter  
net.trainParam.show=m7;  
net.trainParam.lr=m3;  
net.trainParam.mc=m4;  
net.trainParam.epochs=m6;  
net.trainParam.goal=m5;  
-----
```

2. Penentuan kondisi berhenti

Pelatihan jaringan akan berhenti jika informasi *error* jaringan \leq galat *error*.

3. Set pelatihan sebanyak 85 data (pasangan vektor *input* dan vektor target) akan menjalani tahap – tahap seperti berikut:

a. Umpan Maju

Pada tahap ini setiap unit *input* (X_i , $i=1,..6$) menerima sinyal *input* X_i dan menyiarkan sinyal tersebut ke semua unit yang ada pada lapisan atasnya yaitu *hidden layer* pertama.

Unit – unit pada *hidden layer* pertama (Y_m , $m=1,..110$) menjumlahkan sinyal *input* berbobotnya dengan menggunakan persamaan:

$$Y_{net_j} = V_{om} + \sum_{i=1}^{110} x_i V_{ik}$$

kemudian menerapkan fungsi aktivasi untuk menghitung sinyal *output*nya:

$$Y_j = f(Y_{net_j}) = \frac{1}{1 + e^{-Y_{net_j}}}$$

sinyal ini kemudian dikirim ke semua unit lapisan di atasnya (unit – unit *output*).

Unit *output* (Z_o , $o=1$) akan menjumlahkan sinyal *input* dari *hidden layer* pertama dengan menggunakan persamaan:

$$Z_{net_k} = W_{om} + \sum_{j=1}^{10} Y_k W_{ko}$$

dan menerapkan kembali fungsi aktivasi untuk menghitung nilai *output*nya:

$$Z_k = f(Z_net_k) = \frac{1}{1 + e^{-z_net_k}}$$

b. *Error Backpropagation*

Unit *output* Z_o menerima sebuah pola target (t_o) yang berhubungan dengan pola latihan *input*, kemudian persamaan informasi *error*nya (δ_o) dihitung,

$$\begin{aligned} \delta_o &= (t_o - Z_o) f'(Z_net_k) \\ &= (t_o - Z_o) y_o (1 - y_o) \end{aligned}$$

Demikian pula persamaan koreksi bobotnya yang akan digunakan untuk merubah bobot W_{ko} dengan laju pembelajaran α

$$\Delta W_{ko} = \alpha \delta_o Y_k$$

Sedangkan untuk menghitung persamaan koreksi biasnya dengan persamaan seperti berikut:

$$\Delta W_{om} = \alpha \delta_o$$

Kemudian menyiarkan δ_o ke lapisan di bawahnya (δ_o dipropagasikan balik). Persamaan koreksi bias ini nantinya dipergunakan untuk meng-update w_{om} .

Setiap unit tersembunyi hidden layer kedua ($Y_k, k=1, \dots, 110$) menjumlahkan *input* – *input* delatanya (dari unit – unit pada lapisan di atasnya)

$$\delta_in_k = \sum \delta_o W_{ko}$$

mengalikannya dengan turunan dari fungsi aktivasinya untuk menghitung persamaan informasi *error*nya dengan menggunakan formula:

$$\delta_k = \delta_in_k \cdot f'(Y_net_j) = \delta_in_k \cdot (Y_net_j) \cdot (1 - Y_net_j)$$

$$\Delta V_{ik} = \alpha \delta_k X_i$$

$$\Delta V_{om} = \alpha \delta_k$$

c. *Perubahan bobot dan bias*

Tiap – tiap unit *output* ($Z_o, o=1$) akan mengupdate bias dan bobotnya ($m=1, \dots, 110$):

$$W_{ko}(\text{baru}) = W_{ko}(\text{lama}) + \Delta W_{ko}$$

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

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

d. Pelatihan Jaringan

Setelah membangun JST *Backpropagation* dan menentukan parameter jaringan langkah komputasi selanjutnya adalah melakukan proses pelatihan JST. Matlab menyediakan fungsi untuk melakukan proses ini seperti berikut :

```
-----  
%Menjalankan pembelajaran  
net=train(net,A,normal,T);pause  
-----
```

Fungsi di atas dengan menerapkan persamaan (2.12) sampai (2.22) melakukan proses pelatihan jaringan sampai menghasilkan *error* minimum.

Pelatihan jaringan digunakan untuk melatih set data yang telah dibuat, yaitu data *input* berupa variabel faktor lingkungan yang mempengaruhi penyebaran penyakit DBD di wilayah Kabupaten Brebes yang meliputi suhu rata – rata, jumlah hari hujan, curah hujan, kepadatan penduduk, ketinggian wilayah, serta %ABJ. Sedangkan data targetnya adalah kondisi wilayah terhadap kasus penyakit DBD. Pelatihan dilakukan dengan variasi parameter jaringan yaitu jumlah *neuron* pada masing-masing *hidden layer*, laju pembelajaran, 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 laju pembelajaran 0,1, konstanta momentum 0,7, kinerja tujuan (*target error*) 0,001, maksimum iterasi 15000 *epoch* dan dengan algoritma pelatihan *gradient descent* dengan momentum dan laju pembelajaran (*traingdx*).

- **Pelatihan 2, Mencari Momentum dan Laju pembelajaran Terbaik**

Pada langkah ini akan diamati variasi momentum dan laju pembelajaran 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 laju pembelajaran yang optimum. Dengan memakai arsitektur

jaringan terbaik pada pelatihan 1 maka akan diamati pengaruh momentum dan laju pembelajaran terhadap unjuk kerja jaringan yaitu pada proses pelatihan.

Pada pelatihan 2, pengamatan dilakukan secara konstruktif dengan konstanta laju pembelajaran 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 pelatihannya *gradient descent* dengan momentum dan laju pembelajaran (*traingdx*).

e. 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.

3.3. Alat dan Bahan Penelitian

1. Alat

Pada penelitian ini memanfaatkan alat penelitian berupa system computer dan perangkat lunaknya dengan spesifikasi sebagai berikut:

a. Perangkat Keras

Untuk mengembangkan sistem informasi penyebaran penyakit demam berdarah menggunakan metode JST Backpropagation ini digunakan seperangkat komputer dengan spesifikasi sebagai berikut:

- Processor Intel Pentium dual core (T4400)
- RAM 2 GB
- HDD 320 GB

b. Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian sebagai berikut:

- OS : Windows
- Tools : Matlab 7.04

2. Bahan

Tujuan dari penelitian ini adalah menginvestigasi kemungkinan penerapan JST *Backpropagation* untuk sistem informasi penyebaran penyakit DBD berdasarkan parameter *real* berupa data – data faktor lingkungan yang mempengaruhi terjadinya penyebaran penyakit DBD. Menurut keterangan ahli epidemiologi, Bapak Awaludin, SKM, M.Kes selaku Kepala seksi Pengendalian Penyakit Dinas Kesehatan Kabupaten Brebes, data/faktor lingkungan yang mempengaruhi penyebaran penyakit DBD tersebut meliputi (*Lampiran 7*): Suhu rata – rata, Kepadatan penduduk, Curah hujan, Ketinggian daratan/wilayah dari permukaan air laut, Mobilitas penduduk, Serta parameter kepadatan jentik nyamuk *Aedes Aegypti* meliputi HI, CI, dan BI yang terakumulasi dengan persentase Angka Bebas Jentik (ABJ).

Berdasarkan keterangan dari ahli, maka bahan penelitian ini berupa data – data faktor lingkungan yang dapat mempengaruhi penyerbaran penyakit demam berdarah di Kabupaten Brebes. Adapun data – data tersebut meliputi:

- a. Data suhu rata – rata yang bersumber dari BMKG Kota Tegal
- b. Data jumlah hari hujan yang bersumber dari BPS Kabupaten Brebes
- c. Data curah hujan yang bersumber dari BPS Kabupaten Brebes
- d. Data kepadatan penduduk, bersumber dari BPS Kabupaten Brebes
- e. Ketinggian daratan dari Permukaan air laut, bersumber dari BPS Kabupaten Brebes
- f. Data % Angka bebas jentik wilayah Brebes, dari Dinas Kesehatan Kabupaten Brebes
- g. Data kasus demam berdarah wilayah Kabupaten Brebes dari Dinas Kesehatan Kabupaten Brebes.

3.4.Perancangan Analisis Hasil

Kinerja dari suatu JST setelah dilakukan pelatihan dapat diukur dengan melihat *error* hasil pelatihan, dan pengujian terhadap sekumpulan data masukan baru. Hasil pelatihan dan pengujian dapat dianalisis dengan mengamati ketepatan atau akurasi antara target dengan keluaran jaringan. Setelah sistem dilatih, maka langkah selanjutnya adalah validasi sistem. Pada proses validasi, sistem diuji dengan data lain, hal ini dimaksudkan untuk mengetahui sejauh mana sistem dapat menginformasikan nilai – nilai keluaran dari nilai – nilai masukan yang diberikan. Keluaran jaringan akan dibandingkan dengan target data pengujian.

Persamaan (3.2) digunakan untuk memvalidasi sistem JST (Sandra, 2005):

$$\text{Validasi}(\%) = \left[\frac{\text{Jumlah_data_sesuai_target}}{\text{Jumlah_data}} \right] \times 100\% \quad (3.2)$$