

# Jaringan Syaraf Tiruan Untuk Klasifikasi Penyakit Demam Menggunakan Algoritma Backpropagation

Dwi Andini

Fakultas Ilmu Komputer dan Teknologi Informasi, Prodi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia  
Email : [dwiandini279@gmail.com](mailto:dwiandini279@gmail.com)

**Abstrak**—Kemajuan teknologi telah membantu menyelesaikan permasalahan di berbagai bidang terutama bidang kesehatan, salah satunya dalam pengklasifikasi penyakit yang mempermudah manajemen pengawasan penyakit untuk melihat penyakit tersebut masuk dalam kategori jenis penyakit apa. Proses klasifikasi menggunakan computer dapat diterapkan dengan menggunakan berbagai macam metode-metode klasifikasi salah satunya metode Jaringan Syaraf Tiruan dengan Algoritma *Backpropagation*. Jaringan syaraf tiruan merupakan salah satu system pemrosesan informasi yang didesain dengan menirukan cara kerja otak manusia dengan melakukan proses belajar melalui perubahan bobot sinapsisnya. Salah satu permasalahan yang dapat menerapkan algoritma *Backpropagation* dalam kasus klasifikasi adalah klasifikasi Penyakit Demam (Demam Berdarah *Dengeu* dan *Typhoid*) karena kemiripan gejala yang dimiliki kedua penyakit tersebut. Penerapan algoritma *Backpropagation* dalam Klasifikasi Penyakit Demam (Demam Berdarah *Dengeu* dan *Typhoid*) diawali dengan tahapan pelatihan pada data latih sebanyak 135 data, diperoleh variasi *learning rate* dan *neuron hidden layer* terbaik dengan cara *trial* dan *error*. Pengujian dilakukan pada data uji yaitu sebanyak 15 data, hasil pengujian berupa klasifikasi penyakit demam yang dibandingkan hasilnya dengan target sebenarnya.

**Kata Kunci:** *Backpropagation*; Jaringan Syaraf Tiruan; Demam Berdarah *Dengeu* dan *Typhoid*

**Abstract**—Technological advances have helped solve problems in various fields, especially in the health sector, one of which is in disease classification which makes it easier to control disease management to see what type of disease the disease belongs to. The classification process using a computer can be applied using various classification methods, one of which is the Artificial Neural Network method with the *Backpropagation* Algorithm. An artificial neural network is an information processing system that is designed to imitate the workings of the human brain by carrying out the learning process through changes in the weight of its synapses. One of the problems that can apply the *Backpropagation* algorithm in the case of classification is the classification of Fever Disease (*Dengue* and *Typhoid*) because of the similarity of the symptoms of the two diseases. The application of the *Backpropagation* algorithm in the Classification of Fever (*Dengue* and *Typhoid* *Dengue* Fever) begins with the training stages on 135 training data, and the best variation of *learning rate* and *hidden layer neurons* is obtained by trial and error. The test was carried out on test data as many as 15 data, the test results were in the form of a classification of fever diseases which were compared with the actual target.

**Keywords:** *Backpropagation*; Artificial Neural Networks; *Dengue Hemorrhagic Fever* and *Typhoid*.

## 1. PENDAHULUAN

Demam merupakan naiknya suhu tubuh menjadi lebih tinggi dari biasanya. Suhu tubuh normal manusia berada pada titik  $37^{\circ}\text{C}$ , jika suhu tubuh lebih dari angka tersebut menunjukkan adanya demam yang disebabkan oleh faktor infeksi atau faktor non infeksi. Demam merupakan hal yang sering terjadi pada manusia dan merupakan indikator bahwa tubuh sedang melakukan perlawanan terhadap zat-zat berbahaya. Terdapat 8 jenis demam yang perlu diwaspadai antara lain Demam Berdarah *Dengeu*, *Typhoid*, Malaria, *Chicken Guinea*, Viral, *Meningitis*, Infeksi saluran kemih dan HIV. Dari 8 jenis demam tersebut tiga diantaranya memiliki gejala yang mirip yaitu Demam Berdarah *Dengeu*, *Thypoid* dan Malaria. Demam berdarah *dengeu* merupakan penyakit infeksi yang dapat berakibat fatal. Dalam waktu yang relatif singkat, penyakit ini dapat merenggut nyawa penderitanya jika tidak ditangani secepatnya. Menurut *World Health Organization* (WHO 2009), terdapat tiga tahapan yang dialami penderita penyakit DBD, yaitu fase demam, fase kritis, dan fase pemulihan. Menurut WHO tahun 1997, kriteria diagnosis penyakit demam berdarah *dengeu* terdiri dari kriteria klinis dan laboratoris [1].

Kota Medan merupakan salah satu kota dengan jumlah kasus penderita Demam Berdarah *Dengeu* dan *Typhoid* yang cukup tinggi. Salah satu Rumah Sakit di Medan yang menangani pasien penderita penyakit Demam Berdarah *Dengeu* dan *Typhoid* adalah RSU Bakti. Rumah Sakit Umum Bakti adalah Rumah sakit swasta yang berada di JL.HM. Joni No.64, Teladan Tim.,Kec. Medan Kota,Medan, Sumatera Utara. Rumah Sakit Umum Bakti memiliki ketersediaan data penyakit Demam berdarah *dengeu*, *typhoid* sehingga relevan dengan penelitian ini.

Gejala utama pada penyakit Demam berdarah *dengeu* dan *typhid* yaitu demam, demam pada penyakit Demam Berdarah *dengeu* yaitu demam tinggi dengan suhu antara  $30^{\circ}\text{C}$  -  $40^{\circ}\text{C}$  yang muncul secara mendadak, demam berlangsung selama 7 hari dan terjadi secara terus menerus, suhu tubuh bisa normal atau rendah, lalu akan naik secara perlahan setiap hari dan bisa mencapai  $40^{\circ}\text{C}$ . Kedua penyakit ini masih menjadi masalah kesehatan masyarakat di kota Medan. Penyakit Demam berdarah *dengeu* dan *tifoid* memiliki jumlah penderita yang banyak dan memiliki resiko kematian yang cukup tinggi. Kemiripan gejala dari masing-masing penyakit sering menimbulkan kesulitan dalam mendapatkan *anamnese* (diagnosa sementara) karena waktu tunggu yang cukup lama, menyebabkan kecepatan mendiagnosis penyakit sangat terbatas dan terkadang kurang akurat sehingga pasien mendapatkan penanganan awal yang kurang tepat dan semakin memperburuk kondisi pasien [2].

Demam *Typhoid* adalah infeksi akut pada saluran pencernaan yang disebabkan oleh *salmonella typhi*. Demam *Typhoid* merupakan penyakit endemic di Indonesia. Penyakit ini termasuk penyakit menular yang tercantum dalam undang-undang nomor 6 Tahun 1962 tentang wabah. Kelompok penyakit menular ini merupakan penyakit yang mudah menular dan dapat menyerang banyak orang sehingga dapat menimbulkan wabah. Demam *Typhoid* menyerang penduduk di semua Negara. Seperti penyakit menular lainnya, *Tifoid* banyak ditemukan di Negara berkembang di mana *hygiene* pribadi dan sanitasi lingkungannya kurang baik. Prevalensi kasus bervariasi tergantung lokasi, kondisi lingkungan setempat, dan perilaku masyarakat. Angka insiden di seluruh dunia sekitar 17 juta pertahun dengan 600.000 orang meninggal karena penyakit ini. Menurut *World Health Organization* (WHO 2009), terdapat tiga tahapan yang dialami penderita penyakit DBD, yaitu fase demam, fase kritis, dan fase pemulihan. Menurut WHO tahun 1997, kriteria diagnosis penyakit demam berdarah *dengue* terdiri dari kriteria klinis dan laboratoris. *World Health Organization* (WHO) memperkirakan 70% kematian terjadi di Asia. Di Indonesia, insiden demam *tifoid* banyak dijumpai pada populasi yang berusia 3-19 tahun. Menurut kementerian kesehatan tahun 2006, gambaran klinis *Tifoid* sangat bervariasi, dari gejala yang ringan sekali (sehingga tidak terdiagnosis) dan dengan gejala yang khas (sindrom demam *tifoid*) sampai dengan gejala klinis berat yang disertai komplikasi. Gambaran klinis juga bervariasi berdasarkan daerah atau Negara, serta menurut waktu. Gambaran klinis di negara berkembang dapat berbeda dengan negara maju dan gambaran klinis tahun 2000 dapat berbeda dengan tahun sekarang pada daerah yang sama [3].

Jaringan Syaraf Tiruan berkembang secara pesat pada beberapa tahun terakhir. Jaringan Syaraf Tiruan telah dikembangkan sebelum adanya suatu komputer konvensional yang canggih dan terus berkembang walaupun pernah mengalami masa vakum selama beberapa tahun. Jaringan Syaraf Tiruan merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran otak manusia tersebut. Jaringan saraf tiruan (JST), atau juga disebut *simulated neural network* (SNN), atau umumnya hanya disebut *neural network* (NN) merupakan jaringan dari sekelompok unit neuro untuk melakukan prediksi terhadap apa yang terjadi dimasa yang akan datang. Secara sederhana, JST adalah sebuah alat pemodelan data statistik non-linier. JST dapat digunakan untuk memodelkan hubungan yang kompleks antara input dan output untuk menemukan pola-pola pada data. Berdasarkan pendapat ahli di atas, maka Jaringan Syaraf Tiruan merupakan paradigma pemrosesan suatu informasi yang terinspirasi oleh sistem sel syaraf biologi, sama seperti otak yang memproses suatu informasi [4].

Jaringan syaraf tiruan pada saat ini sangat dibutuhkan untuk menjadikan alat untuk *cloning* data maupun cara berfikir seseorang dengan menerapkan metode yang berkaitan dengan jaringan saraf tiruan. Otak manusia terdiri atas jutaan neuron yang saling terhubung yang dikenal sebagai syaraf biologi. Setiap neuron terdiri atas sebuah sel yang memiliki sejumlah dendrit (*input*) dan sebuah akson (*output*). Akson berhubungan dengan neuron yang lain melalui jalur penghubung yang menghasilkan reaksi kimia saat merespon *input* yang masuk. Salah satu algoritma Jaringan Syaraf Tiruan yang sangat cocok diterapkan dalam kasus klasifikasi adalah algoritma *Backpropagation*. Kelebihan yang dimiliki *Backpropagation* adalah melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa dengan pola yang dipakai selama proses pelatihan [5].

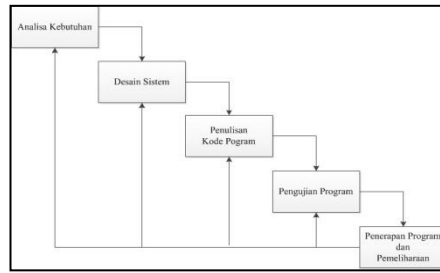
*Backpropagation* adalah salah satu algoritma pada jaringan syaraf tiruan yang sering digunakan dalam mencari bobot optimal. Pada jaringan *Backpropagation* terdapat pola input dan pola output yang diinginkan. Pada saat jaringan diberikan suatu pola, nilai bobot-bobot diubah agar dapat memperkecil perbedaan antara pola output dari jaringan dan pola output yang diinginkan. Pelatihan jaringan dilakukan berulang-ulang sampai semua pola output dari jaringan dapat mengenali pola output yang diinginkan [6].

## 2. METODOLOGI PENELITIAN

### 2.1 Kerangka Kerja Penelitian

Kerangka kerja penelitian merupakan suatu tahapan yang digunakan dalam langkah-langkah penyelesaian masalah yang akan dilakukan dalam penelitian. Kerangka kerja penelitian merupakan rencana yang memuat garis besar dari suatu karangan yang akan diteliti, dan merupakan ide-ide yang disusun secara sistematis, logis, jelas, terstruktur dan teratur. Kerangka kerja penelitian ini membahas tentang cara atau langkah-langkah yang sistematis dan juga mencakup prosedur dan teknik penelitian yang bisa memecahkan masalah-masalah yang ada [7].

Penelitian ini juga membahas topik tentang “Jaringan Syaraf Tiruan Untuk Klasifikasi Penyakit Demam Menggunakan Algoritma *Backpropagation*”. Dalam penelitian yang dilakukan, ada beberapa tahapan-tahapan kerangka kerja penelitian yang digunakan adalah:



**Gambar 1.** Skema Metode Waterfall

Metode *waterfall* merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Inti dari metode *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi jika langkah ke-1 belum dikerjakan, maka langkah 2 tidak dapat dikerjakan. Jika langkah ke-2 belum dikerjakan maka langkah ke-3 juga tidak dapat dikerjakan, begitu seterusnya. Secara otomatis langkah ke-3 akan bisa dilakukan jika langkah ke-1 dan ke-2 sudah dilakukan. Secara garis besar metode *waterfall* mempunyai langkah-langkah sebagai berikut: Analisa, Desain, Penulisan, Pengujian dan Penerapan serta Pemeliharaan.

Dalam melakukan penelitian ini dengan menggunakan metode *waterfall* maka beberapa uraian kerangka kerja yang harus dilakukan terlebih dahulu untuk dapat memproses data penyakit demam pada pasien melalui langkah-langkah sebagai berikut:

- a. **Analisa Kebutuhan**  
Analisa kebutuhan berfungsi untuk melakukan analisa data yang akan diolah kedalam system jaringan saraf tiruan, seperti menentukan gejala demam yang terjadi kepada pasien, menentukan system dan metode yang digunakan dalam penelitian. Sistem analis akan menggali informasi sebanyak-banyaknya dari *user* sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas seseorang sebelumnya. Tahapan ini akan menghasilkan dokumen *user requirment* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan sistem.
- b. **Desain Sistem**  
Desain system berfungsi untuk melakukan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem seperti diagram alir data (*data flow diagram*), diagram hubungan entitas (*entity relationship diagram*) serta struktur dan bahasan data.
- c. **Penulisan Kode Program**  
Penulisan kode program atau *coding* merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang akan meterjemahkan transaksi yang diminta oleh *user*. Tahapan ini lah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.
- d. **Pengujian Program**  
Tahapan akhir dimana sistem yang baru diuji kemampuan dan keefektifannya sehingga didapatkan kekurangan dan kelemahan sistem yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi menjadi lebih baik dan sempurna.
- e. **Penerapan Program dan Pemeliharaan**  
Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (periperal atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

## 2.2 Jaringan Syaraf Tiruan (*Artificial Neural Networks*)

Jaringan Saraf Tiruan (JST) saat ini telah berkembang dengan pesat dan telah diimplementasikan dalam berbagai bidang. Salah satu implementasi JST adalah dalam bidang pendidikan untuk melakukan prediksi kelulusan sidang skripsi. Prediksi kelulusan sidang skripsi telah menjadi salah satu topik yang menarik, mengingat keuntungan yang diperoleh apabila dapat melakukan prediksi kelulusan sidang skripsi dengan baik[1]. Jaringan Syaraf Tiruan adalah merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran otak manusia tersebut [2].

## 2.3 Metode Backpropagation

*Backpropagation* adalah salah satu algoritma pada jaringan syaraf tiruan yang sering digunakan dalam mencari bobot optimal. Pada jaringan *Backpropagation* terdapat pola input dan pola output yang diinginkan. Pada saat jaringan diberikan suatu pola, nilai bobot-bobot diubah agar dapat memperkecil perbedaan antara pola output dari jaringan dan pola output yang diinginkan, Pelatihan jaringan dilakukan berulang-ulang sampai semua pola output dari jaringan dapat mengenali pola output yang diinginkan [4].

### 2.3.1 Inisialisasi Bobot Awal

Inisialisasi bobot awal dari algoritma ini diperoleh dari proses acak atau dengan menggunakan algoritma Nguyen-Widrow. Algoritma Nguyen-Widrow merupakan algoritma optimasi nilai bobot dan bias awal dari jaringan *Backpropagation* sehingga proses belajar jaringan untuk mencapai bobot optimal menjadi lebih cepat dibandingkan dengan penginisialisasian secara random/acak. Algoritma inisialisasi Nguyen-Widrow adalah sebagai berikut: [5]

- Tentukan:  $n$  = jumlah neuron input  $p$  = jumlah neuron pada lapisan tersembunyi  $\beta$  = faktor skala dengan nilai  $n$   $0,7 p$
- Inisialisasi bobot-bobot ( $ij$  v (lama)) menggunakan nilai acak antara -0,5 dan 0,5.
- Hitung  $2 \cdot 2 \cdot 2 \cdot ij \cdot 1 \cdot j \cdot nj \cdot v = v + v + K + v \cdot 4$ . Bobot yang dipakai sebagai inisialisasi  $ij \cdot v \cdot v$  lama  $v \cdot \beta$  ( ) = Nilai bias yang digunakan,  $v0 \cdot j$  = bilangan antara  $-\beta$  dan  $\beta$

### 2.3.2 Fungsi Aktivasi

Fungsi aktivasi yang digunakan dalam *Backpropagation* harus memenuhi tiga syarat, yaitu kontinu, terdeferensial dengan mudah, dan merupakan fungsi tidak turun.

Fungsi aktivasi yang paling sering digunakan dalam jaringan syaraf tiruan adalah fungsi sigmoid biner dan didefinisikan sebagai berikut.  $( ) x \in f x - + = 1$  dan turunan  $f'(x) = f(x)(1-f(x))$  Fungsi sigmoid biner mempunyai nilai maksimum satu. Sehingga untuk pola input dan output dengan nilai lebih besar dari satu harus ditransformasi agar setiap polanya memiliki range nilai yang sama dengan fungsi sigmoid biner.

### 2.3.3 Algoritma Pelatihan *Backpropagation*

Berikut adalah algoritma pelatihan *Backpropagation*: [5]

- Inisialisasi bobot awal dengan bilangan acak kecil.
- Tentukan maksimum iterasi, Target error yang diinginkan, dan Learning rate ( $\alpha$ ).
- Bila kondisi berhenti bernilai FALSE, kerjakan langkah-langkah berikut:

Untuk setiap pasangan elemen pelatihan, kerjakan:

Feedforward:

- Setiap neuron input  $i$   $x$  dengan  $i = 1, 2, 3, K, n$ , mendapat sinyal  $i$   $x$  dan diteruskan ke semua neuron pada lapisan tersembunyi.
- Setiap neuron pada suatu lapisan tersembunyi  $j$   $z$  dengan  $j = 1, 2, 3, K, p$ , menjumlahkan sinyal-sinyal input berbobot:  $\sum = + n \cdot i \cdot j \cdot i \cdot ij \cdot z \cdot net \cdot v \cdot x \cdot v \cdot 1 \cdot 0$  dengan menerapkan fungsi aktivasi, hitung sinyal output pada lapisan tersembunyi:  $( ) j \cdot net \cdot j \cdot z = f \cdot z$  selanjutnya kirimkan sinyal tersebut ke semua neuron di lapisan output.
- Setiap neuron output  $k$   $y$  dengan  $k = 1, 2, 3, K, m$ , menjumlahkan sinyal-sinyal input berbobot:  $\sum = + p \cdot j \cdot k \cdot k \cdot jwk \cdot y \cdot net \cdot w \cdot z \cdot 1 \cdot 0$  dengan menerapkan fungsi aktivasi, hitung sinyal output pada lapisan output:  $( ) k \cdot net \cdot k \cdot y = f \cdot y$

Catatan: langkah (b) dilakukan sebanyak jumlah lapisan tersembunyi.

*Backpropagation*:

- Setiap neuron output  $k$   $y$  dengan  $k = 1, 2, 3, K, m$ , menerima target pola output yang berkaitan dengan pola input pelatihan. Hitung error pada lapisan output:  $( ) ( ) k \cdot k \cdot k \cdot net \cdot k \cdot \delta = t - y \cdot f' \cdot y \cdot k \cdot t$  adalah target output yang diharapkan. Selanjutnya hitung perubahan bobot (digunakan untuk memperbaiki nilai bobot  $wjk$ ):  $j \cdot k \cdot j \cdot k \cdot \Delta w = \alpha \delta \cdot z$  Hitung juga perubahan bobot bias (digunakan untuk memperbaiki nilai bobot  $w0k$ ):  $\Delta w0k = \alpha \delta \cdot k$  Kirimkan  $\delta \cdot k$  ini ke neuron pada lapisan tersembunyi.
- Setiap neuron pada lapisan tersembunyi  $j$   $z$  dengan  $j = 1, 2, 3, K, p$ , menjumlahkan faktor delta pada lapisan tersembunyi:  $\sum = m \cdot k \cdot net \cdot j \cdot kwjk \cdot 1 \cdot \delta \cdot \delta$  nilai ini dikalikan dengan turunan dari fungsi aktivasinya dan digunakan untuk menghitung informasi error pada lapisan tersembunyi:  $( )$  selanjutnya hitung perubahan bobot (digunakan untuk memperbaiki nilai bobot  $ij \cdot v$ ):  $ij \cdot j \cdot i \cdot \Delta v = \alpha \delta \cdot x$  Selanjutnya hitung perubahan bobot bias (digunakan untuk memperbaiki nilai bobot  $j \cdot v0$ ):  $j \cdot j \cdot \Delta v0 = \alpha \delta$  Perubahan bobot.
- Setiap neuron output  $k$   $y$  dengan  $k = 1, 2, 3, K, m$ , memperbaharui bobot dan biasnya ( $j = 0, 1, 2, K, , p$ ):  $wjk \cdot * = wjk + \Delta wjk$  setiap neuron pada lapisan tersembunyi  $j$   $z$  dengan, memperbaharui bobot dan biasnya ( $i = 0, 1, 2, K, n$ ):  $ij \cdot ij \cdot v \cdot * = v + \Delta v$ .

Kondisi penghentian yang digunakan dalam algoritma ini adalah jumlah maksimum iterasi dan target error. Pelatihan akan dihentikan apabila jumlah iterasi pelatihan melebihi jumlah iterasi maksimum atau jika error yang diperoleh dalam pelatihan lebih kecil dari target error. Pada tahap perubahan bobot dapat dimodifikasi dengan menambahkan momentum ( $\mu$ ) agar lebih cepat mencapai kekonvergenan. Kecepatan kovergensi juga dapat ditingkatkan dengan memodifikasi learning rate menjadi adaptive dimana learning rate akan berubah-ubah selama proses pelatihan. Jika selisih antara error yang diperoleh dan error sebelumnya melebihi nilai maksimum kenaikan kinerja maka nilai bobot-bobot yang baru diabaikan, dan nilai leraning rate diturunkan dengan cara mengalikan dengan nilai penurunan leraning rate. Sebaliknya jika selisih error yang diperoleh dan error sebelumnya kurang dari maksimum kenaikan kinerja, semua nilai bobot yang baru akan disimpan, dan nilai learning rate ditingkatkan dengan cara mengalikan dengan nilai kenaikan learning rate.

Setelah pelatihan, dilakukan pengujian jaringan *Backpropagation* dengan hanya menggunakan tahap feedforward dengan menggunakan bobot yang diperoleh dari proses pelatihan.

### 2.3.4 Pengukuran Ketepatan Model Prakiraan

Mengukur ketepatan model prakiraan dapat dihitung dengan persamaan berikut:

$$\text{Akurasi} = \frac{\text{Total Prakiraan yang benar} \times 100\%}{\text{Total Prakiraan}}$$

## 2.4 Demam

Demam merupakan naiknya suhu tubuh menjadi lebih tinggi daripada biasanya. Suhu tubuh normal manusia berada pada titik  $37^{\circ}\text{C}$ , jika tubuh menunjukkan lebih dari angka tersebut menunjukkan adanya demam yang disebabkan oleh infeksi. Demam juga merupakan ciri dari sel antibody manusia sedang melawan virus atau bakteri. Terdapat 8 jenis demam yang perlu diwaspadai Antara lain, Demam Berdarah Dengeu, Tifoid, Malaria, *Chicken Guinea*, Viral, Meningitis, Infeksi saluran kemih dan HIV. Dari 8 jenis demam tersebut 2 diantaranya memiliki gejala yang mirip yaitu Demam Berdarah *Dengeu* dan *Tifoid* [6].

## 3. HASIL DAN PEMBAHASAN

### 3.1 Analisa Masalah

Permasalahan terhadap sistem yang dibangun tentang mengklasifikasikan penyakit demam berdarah di Rumah Sakit Bakti Medan, akan dilakukan dengan melakukan analisa data terlebih dahulu, setelah data ditentukan lalu menerapkan metode *backpropagation* pada sistem yang akan dibangun, di mana sistem yang dibangun dengan menerapkan jaringan saraf tiruan. Pada permasalahan yang ada, terdapat 3 lapisan yang terdiri dari lapisan masukan (layer input) di mana jumlah neuron pada lapisan input sebanyak 5 yang ditentukan pada jumlah gejala penyakit, satu lapisan tersembunyi. Untuk menentukan jumlah neuron pada hidden layer dan lapisan output untuk menentukan hasil klasifikasi. Adapun rumus yang digunakan dalam menentukan lapisan pada jaringan saraf tiruan sebagai berikut:

$$N_h = \sqrt{N_i * N_o}$$

Keterangan:

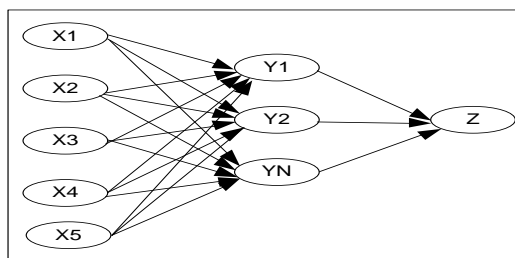
$N_h$  = Jumlah Neuron Hidden

$N_i$  = Jumlah Neuron Input

$N_o$  = Jumlah Neuron Output

Misalkan jumlah neuron pada layer input sebanyak 5 dan jumlah neuron pada layer output sebanyak 2 maka Neuron hidden dapat dihitung sebagai berikut:  $N_h = \sqrt{5 * 2}$  Atau  $N_h = \sqrt{10} = 3.16$  dibulatkan kebawah menjadi 3.

Model jaringan syaraf tiruan dalam memprediksi demam menggunakan metode Backpropagation dengan menggunakan struktur jaringan syaraf tiruan 5 layer/lapisan. Pada system yang akan dirancang dalam menentukan klasifikasi penyakit demam pada rumah sakit Bakti medan, akan menerapkan struktur jaringan syaraf tiruan yang terdiri dari 5 layer . dimana setiap layer memiliki fungsi masing-masing seperti 5 layer input yang digunakan untuk mengklasifikasikan penyakit dari gejala yang timbul, 3 layer hidden dan 1 layer output. Pada penelitian ini masing-masing layer mempunyai net struktur 5- 3- 1 (5 unit layer input, 3 unit layer hidden, 1 unit layer output) seperti pada gambar dibawah ini.



**Gambar 3.** Struktur Jaringan syaraf tiruan

Keterangan:

$X_1$ - $X_2$  = Input Gejala yang diderita pasien

$Y_1$ - $Y_n$  = Banyaknya Layer Tersembunyi (Hidden Layer)

$Z$  = Klasifikasi jenis penyakit

Algoritma *Backpropagation* merupakan pembelajaran tersupervisi dan terutama digunakan oleh Multi-layer-Neural Networks untuk mengubah bobot yang terhubung dengan neuron layer tersembunyi pada jaringan. Algoritma Backpropagation menggunakan hitungan error output untuk mengubah nilai bobot dalam arah mundur.

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

Algoritma bekerja sebagai berikut:

- Lakukan fasa Backpropagation untuk pola input dan hitung error output.
- Ubah semua nilai bobot dari matriks bobot dengan menggunakan rumus: bobot lama + learning rate \* error output \* output(neuroni) \* output(neuroni+1) \* (1 – output(neuroni+1)).
- Kembali ke langkah 1.
- Algoritma berakhir, jika semua pola output cocok dengan pola tujuan.

Proses perhitungan deteksi jenis penyakit demam ini dilakukan setelah perolehan data-data gejala klinis dari pakar Demam. Masukan sistem atau input yang dapat dimengerti pada setiap variabel yang mampu menghasilkan diagnosa jenis penyakit Demam yang harus diinisialisasikan terlebih dahulu, adapun variabel yang menandakan bahwa seseorang mengalami penyakit demam seperti pada berikut ini:

- Merasakan sakit kepala
- Badan terasa menggigil
- Muntah
- Demam
- Bercak Darah

Untuk melakukan klasifikasi terhadap proses diagnosa Demam berdarah di rumah sakit Bakti Medan adapun beberapa langkah diantaranya dilakukan adalah mengidentifikasi gejala subjektif dan gejala objektif. Dimana gejala subjektif adalah suatu gejala yang bersifat umum yang memiliki nilai pasti antara “YA” atau “TIDAK”. Untuk variabel yang nilainya tidak bersifat pasti maka nilai variabel ini akan dilakukan proses fuzzyfikasi sehingga output dari nilai variabel ini menghasilkan interval antara 0 sampai 1 . untuk menentukan nilai dari masing-masing variabel , maka dapat dibuat suatu tabel dimana data yang akan diolah akan ditampilkan seperti pada tabel berikut:

**Tabel 2.** Nilai Masing-masing variabel

| No | Variabel-Variabel        | Nilai dari Variabel |
|----|--------------------------|---------------------|
| 1  | Merasa Sakit Kepala (X1) | Tidak = 0           |
|    |                          | Ya = 1              |
| 2  | Badan Menggigil (X2)     | Tidak = 0           |
|    |                          | Ya = 1              |
| 3  | Muntah (X3)              | Tidak = 0           |
|    |                          | Ya = 1              |
| 4  | Demam (X4)               | Tidak = 0           |
|    |                          | Ya = 1              |
| 5  | Adanya Bercak Darah (X5) | Tidak = 0           |
|    |                          | Ya = 1              |

### 3.1 Algoritma Backpropagation

Pada fase ini dilakukan proses adaptasi bobot untuk masing-masing bobot antara layer input dan layer hidden, layer hidden dan layer output. Adaptasi bobot dilakukan secara terus menerus sampai dicapai keadaan error yang paling minimum adapun perhitungan untuk fase pembelajaran adalah bobot awal yang telah di inisialisasi, kemudian dilakukan perhitungan seperti berikut: Hitung harga aktivasi sel-sel pada layer hidden dengan persamaan dibawah ini.

$$Z_j = f(V_{0j} + \sum_i^n X_i V_{ij})$$

Dimana:

$V_{ij}$  adalah sel-sel pada layer input,

$Z_j$  adalah sel pada layer hidden

dan  $X_i$  adalah bobot interkoneksi antara layer input dengan layer hidden, selanjutnya hitung harga aktivasi sel-sel pada layer output dengan persamaan dibawah ini:

$$Y_k = f(W_{0k} + \sum_j W_{kj} * Z_j)$$

Dimana  $Y_k$  adalah sel-sel pada layer output dan  $W_{kj}$  adalah bobot interkoneksi antara layer hidden dengan layer output. Proses perhitungan harga aktivasi ini berlangsung terus sampai pola berakhir. Kemudian hitung harga  $\delta_k$  yang akan digunakan untuk menghitung interkoneksi antara layer hidden dengan layer output dalam jaringan.

**Tabel 3.** Data Klasifikasi Penyakit Demam

| Input |   | Output |
|-------|---|--------|
| 0     | 0 | 0      |
| 0     | 1 | 1      |
| 1     | 0 | 1      |

| 1 | 1 | 0 |
|---|---|---|
|---|---|---|

Keterangan: Input nilai penyakit Demam (1 untuk Demam, 0 untuk Bukan Demam).

Sebelum pelatihan, harus ditentukan terlebih dahulu stopping conditionnya. Misalnya dihentikan jika error telah mencapai 0,41.

**Tabel 4.** Inisiasi bobot dan bias

| No | Inisialisais | Nilai Bobot |
|----|--------------|-------------|
| 1  | V01          | 1,718946    |
| 2  | V11          | -1,263178   |
| 3  | V21          | -1,083092   |
| 4  | W01          | -0,541180   |
| 5  | W11          | 0,543960    |

Step 1: Dengan bobot di atas, tentukan error untuk training data secara keseluruhan dengan Mean Square Error:

$$z_{in11} = 1,718946 + \{(0x - 1,263178) + (0x - 1,083092)\} = 1,718946$$

$$z11 = f(z_{in11}) = 0,847993$$

$$z_{in12} = 1,718946 + \{(0x - 1,263178) + (1x - 1,083092)\} = 0,635854$$

$$z12 = f(z_{in12}) = 0,653816$$

$$z_{in13} = 1,718946 + \{(1x - 1,263178) + (0x - 1,083092)\} = 0,455768$$

$$z13 = f(z_{in13}) = 0,612009$$

$$z_{in14} = 1,718946 + \{(1x - 1,263178) + (1x - 1,083092)\} = -0,627324$$

$$z14 = f(z_{in14}) = 0,348118$$

di mana indeks zjn berarti hidden unit ke – j dan training data ke – n.

$$y_{in11} = -0,541180 + (0,847993 \times 0,543960) = 0,079906$$

$$y11 = f(y_{in11}) = 0,480034$$

$$y_{in12} = -0,541180 + (0,653816 \times 0,543960) = -0,185530$$

$$y12 = f(y_{in12}) = 0,453750$$

$$y_{in13} = -0,541180 + (0,612009 \times 0,543960) = 0,208271$$

$$y13 = f(y_{in13}) = 0,448119$$

$$y_{in14} = -0,541180 + (0,348118 \times 0,543960) = -0,351818$$

$$y14 = f(y_{in14}) = 0,412941$$

$$\text{Maka } E = 0,5 \times \{(0 - 0,480034)^2 + (1 - 0,453750)^2 + (1 - 0,448119)^2 + (0 - 0,412941)^2\} = 0,501957$$

Step2. Karena error masih lebih besar dari 0,41 maka step 3 – 8 dijalankan.

Step 3.  $x1 = 0$ ;  $x2 = 0$  (iterasi pertama, training data pertama)

Step 4.

$$z_{in1} = 1,718946 + \{(0x - 1,263126) + (0x - 1,083049)\} = 1,718946.$$

$$z1 = f(z_{in1}) = 0,847993$$

Step 5.

$$y_{in11} = -0,541180 + (0,847993 \times 0,543960) = 0,079906$$

$$y11 = f(y_{in11}) = 0,480034$$

Step 6.

$$\delta_1 = (0 - 0,480034) f'(0,079906) = -0,119817$$

$$\Delta w11 = 0,01x - 0,119817 \times 0,847993 = -0,001016$$

$$\Delta w01 = 0,01x - 0,119817 = -0,00119817$$

Step 7.

$$\delta_{in1} = -0,00119817 \times 0,543960 = -0,00065176$$

$$\delta_1 = -0,00065176 \times f(1,718946) = -0,00008401$$

$$\Delta v11 = 0,01x - 0,00008401 \times 0 = 0$$

$$\Delta v21 = 0,01x - 0,00008401 \times 0 = 0$$

$$\Delta v01 = 0,01x - 0,00008401 = -0,0000008401$$

Step 8.

$$w01(\text{baru}) = -0,541180 + (-0,00119817) = -0,542378$$

$$w11(\text{baru}) = 0,543960 + (-0,001016) = 0,542944$$

$$v01(\text{baru}) = 1,718946 + (-0,0000008401) = 1,718862$$

$$v11(\text{baru}) = -1,263178 + 0 = -1,263178$$

$$v_{21}(\text{baru}) = -1,083092 + 0 = -1,083092$$

Saat ini  $v_{11}$  dan  $v_{12}$  masih belum berubah karena kedua inputnya = 0. Nilai  $v_{01}$  dan  $v_{02}$  baru berubah pada iterasi pertama untuk training data yang kedua. Setelah step 3-8 untuk training data pertama dijalankan, selanjutnya kembali lagi ke step 3 untuk training data yang kedua ( $x_1=0$  dan  $x_2=1$ ). Langkah yang sama dilakukan sampai pada training data yang keempat. Bobot yang dihasilkan pada iterasi pertama, training data ke-2,3, dan 4 adalah:

**Tabel 5.** Hasil Training

| No | Training | Data Ke-1 | Data Ke-2 | Data Ke-3 | Data Ke-4 |
|----|----------|-----------|-----------|-----------|-----------|
| 1  | W01      | -0,542378 | -0,541023 | -0,539659 | -0,540661 |
| 2  | W11      | 0,542944  | 0,543830  | 0,544665  | 0,544316  |
| 3  | V01      | 1,718862  | 1,718862  | 1,719205  | 1,719081  |
| 4  | V11      | -1,263178 | -1,263178 | -1,263002 | -1,263126 |
| 5  | V21      | -1,083092 | -1,083092 | -1,082925 | -1,083049 |

Setelah sampai pada training data ke-4, maka iterasi pertama selesai. Berikutnya, pelatihan sampai pada step 9, yaitu memeriksa stopping condition dan kembali pada step 2. Demikian seterusnya sampai stopping condition yang ditentukan terpenuhi. Setelah pelatihan selesai, bobot yang didapatkan adalah:

$$v_{01} = 12,719601$$

$$v_{11} = -6,779127$$

$$v_{21} = -6,779127$$

$$w_{01} = -5,018457$$

$$w_{11} = 5,719889$$

Jika ada input baru, misalnya  $x_1=0,2$  dan  $x_2=0,9$  maka outputnya dapat dicari dengan langkah umpan maju sebagai berikut:

Step 0. Bobot yang dipakai adalah bobot hasil pelatihan di atas.

Step 1. Perhitungan dilakukan pada step 2-4

Step 2. Dalam contoh ini, bilangan telah berada dalam interval 0 sampai dengan 1, jadi tidak perlu diskalakan lagi.

Step 3.  $z_{in1} = 12,719601 + \{(0,2 \times -6,779127) + (0,9 \times -6,779127)\} = 5,262561$

$$z_1 = f(5,262561) = 0,994845$$

Step 4.  $y_{ixn1} = -5,018457 + (0,994845 \times 5,719889) = 0,671944$

$$y_1 = f(0,671944) = 0,661938$$

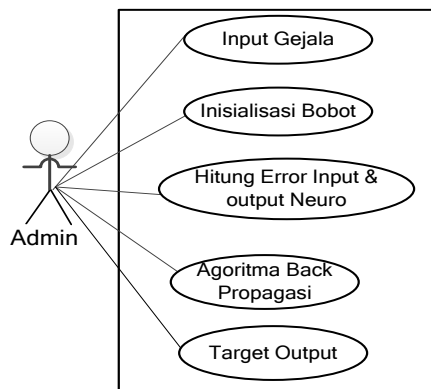
Jadi jika input  $x_1=0,2$  dan  $x_2=0,9$ ; output yang dihasilkan jaringan adalah 0,661938

### 3.2 Perancangan dan Pemodelan

Pemodelan / perancangan sistem merupakan desain awal dalam merancang aplikasi yang akan dibangun dengan pemodelan sistem UML, adapun pemodelan sistem yang akan dibangun seperti berikut ini:

a. Usecase diagram

Diagram *use case* menyajikan interaksi antara *use case* dan aktor. Dimana, aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai. Di mana pada gambar dibawah ini, administrator berfungsi untuk mengendalikan sistem dengan memproses data gejala demam, menginput data bobot setelah itu melakukan analisis dengan metode Backpropagation untuk penentuan nilai Fuzzy Neural Networks yang dapat memprediksikan penyakit demam berdasarkan data target jika 1= Mengalami Penyakit Demam dan 0= Bukan Penyakit Demam, sedangkan user memberikan informasi data bobot dan mendapatkan hasil perhitungan dengan *Backpropagation*.

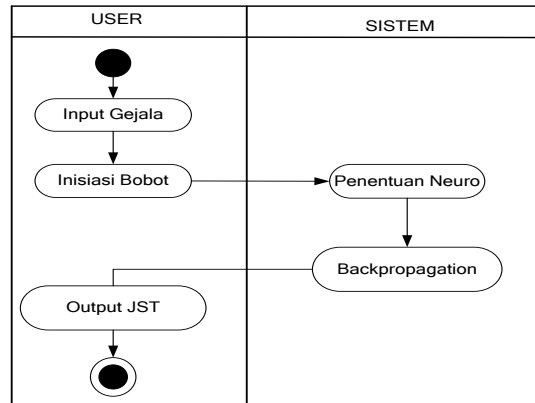


**Gambar 4.** Use Case Diagram Memprediksi Penyakit Demam



b. *Activity Diagram*

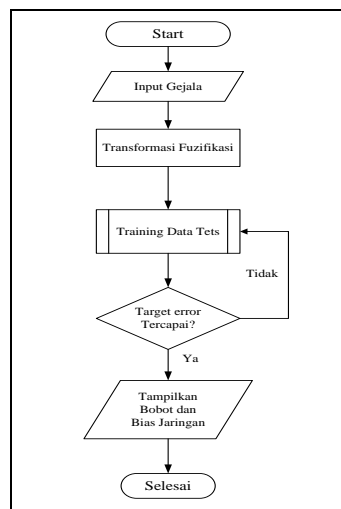
Pada gambar *Activiti Diagram* dibawah menunjukan bahwa user mengendalikan semua sistem yang telah dibuat, dimana user pertama-tama memproses data gejala penyakit demam yang terjadi selama 1 Tahun yang akan diinput kedalam sistem, setelah itu dilakukan pengimputan nilai bobot setiap kriteria yang telah ditentukan maka sistem melakukan kalkulasi dengan perhitungan dengan Backpropagation untuk menentukan hasil prediksi penyakit demam yang dialami pasien, gambar activity diagram dapat dilihat pada gambar dibawah ini:



**Gambar 5.** Activity Diagram Memprediksi Penyakit Demam

2. Flowchart Program

Flowchart program merupakan keterangan yang lebih rinci tentang bagaimana sebenarnya prosedur yang dilakukan oleh suatu program. Flowchart ini menggambarkan urutan logika dari suatu prosedur pemecahan masalah. Adapun flowchart program untuk menentukan harga kamar maksimum seperti yang terlihat pada gambar dibawah ini.

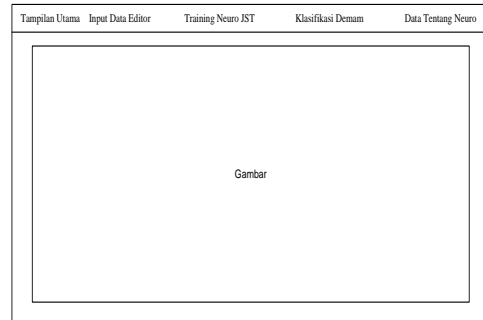


**Gambar 6.** Flowcart Diagram JST Klasifikasi DBD

Untuk memperoleh output yang digunakan maka dibutuhkan beberapa *input* untuk diproses. Adapun form *input* yang telah dirancang untuk Klasifikasi penyakit demam sebagai berikut:

1. Perancangan Input

Perancangan input merupakan awal dimulainya proses informasi. bahan mentah dari informasi adalah data yang terjadi dari transaksi transaksi yang dilakukan oleh organisasi. Perancangan perangkat lunak secara keseluruhan untuk klasifikasi penyakit demam di rumah sakti Bakti, di jalan H.M. Jhoni Medan. Dimana pada form jaringan saraf tiruan metode Backpropagation dalam klasifikasi penyakit demam maka memiliki beberapa parameter seperti jumlah sel lapisan masuk dan lapisan keluar lalu epoch dan selang tampilan, dan pada input nilai diberikan bobot, Form jaringan saraf tiruan yang akan dirancang untuk klasifikasi penyakit demam pada rumah sakit Bakti di jalan H.M Jhoni Medan seperti pada gambar dibawah ini:



**Gambar 7.** Tampilan Utama JST

## 2. Form Tampilan Input Data Editor

**Gambar 8.** Tampilan Input Data Editor

- Button Load File Lain* berfungsi untuk mengupload file training kedalam sistem.
- Text Box* berfungsi untuk menampilkan data training yang diupload.
- Text Box* berfungsi untuk menampilkan data training yang diupload.
- Button Buat File Baru* berfungsi untuk mengimport file baru kedalam sistem atau mengedit data yang terbaru.
- Button Simpan Perubahan* berfungsi untuk menyimpan data perubahan kedalam sistem.

## 3. Tampilan Neuro JST

**Gambar 9.** Tampilan Neuro JST

- Text Box Banyak Neuro pada Hidde Layer* berfungsi untuk mengisi berapa banyak variable atau neuro yang dibutuhkan.
  - Text Box Lama masa pembelajaran dari neuro* berfungsi untuk menginput nilai neuronya.
  - Text Box Nama Input File* berfungsi untuk menampilkan buffering data pada saat dilakukan load file kedalam sistem.
  - Text Box Learning Rate* berfungsi untuk menginput bayaknya lapisan neuro.
  - Button Load file lain* berfungsi untuk mengimport file baru kedalam sistem.
  - Button Progres* berfungsi untuk menampilkan proses upload file pada sistem.
  - Button Reset* berfungsi untuk mereset data yang diupload.
  - Button lakukan training* berfungsi untuk menampilkan hasil training kedalam form.
4. Tampilan Proses Klasifikasi Penyakit.

**Gambar 10.** Tampilan Uji Prediksi

- a. *Text Box* Bulan Tahun berfungsi untuk menginput bulan dan tahun pada klasifikasi penyakit demam.
- b. *Text Box* Merasa sakit kepala berfungsi untuk memilih apakah pasien mengalami gejala sakit kepala atau tidak.
- c. *Text Box* Badan Menggigil berfungsi untuk memilih apakah pasien mengalami gejala Badan Menggigil atau tidak.
- d. *Text Box* Merasa Muntah berfungsi untuk memilih apakah pasien mengalami gejala Muntah atau tidak.
- e. *Text Box* Merasa Demam berfungsi untuk memilih apakah pasien mengalami gejala Demam atau tidak.
- f. *Text Box* ada bercah darah berfungsi untuk memilih apakah pasien mengalami gejala ada bercah darah atau tidak.
- g. *Text Box* klasifikasi penyakit berfungsi untuk menampilkan penyakit pasien, apakah demam atau tidak
- h. *Button* proses berfungsi untuk memproses data dengan metode backpropagation
- i. *Button* kalkulasi hasil nilai berfungsi untuk memproses data dengan jaringan saraf tiruan untuk klasifikasi penyakit demam.
- j. *Button* Reset berfungsi untuk mereset data yang diupload.

## 5. Tampilan Data Tentang Neuro

**Gambar 11.** Tampilan Data Neuron

## 3.3 Implementasi

Untuk dapat menjalankan aplikasi jaringan saraf tiruan dalam mengklasifikasikan penyakit demam dibutuhkan beberapa kebutuhan system, Ada tiga komponen sistem yang dibutuhkan yaitu perangkat lunak(*software*), perangkat keras (*hardware*), dan pengguna (*brainware*).

### 1. Perangkat lunak (*software*)

Perangkat lunak adalah suatu perangkat yang berisi serangkaian instruksi, program, prosedur, pengendali, pendukung, dan aktifitas-aktifitas pengolahan perintah pada sistem komputer. Jadi *software* merupakan komponen abstrak dari susunan sistem komputer. Tanpa *software*, komputer adalah rongsokan elektronik, jadi komputer adalah susunan atas *hardware* dan *software* yang saling bekerjasama. *Hardware* komputer akan hidup dan memiliki fungsi jika digunakan bersama-sama dengan *software*-nya. Antivirus untuk mencegah kemungkinan terjadinya kerusakan sistem yang disebabkan oleh virus-virus yang akan merusak sistem.

### 2. Perangkat keras (*hardware*)

*Hardware* atau dalam bahasa indonesia-nya disebut juga dengan nama “perangkat keras” adalah salah satu komponen dari sebuah komputer yang sifat alat nya bisa dilihat dan diraba secara langsung atau yang berbentuk nyata, yang berfungsi untuk mendukung proses komputerisasi. Hardware dapat bekerja berdasarkan perintah yang telah ditentukan ada padanya, atau yang juga disebut dengan istilah instruction set. Dengan adanya perintah yang dapat dimengerti oleh hardware tersebut, maka hardware tersebut dapat melakukan berbagai kegiatan yang telah ditentukan oleh pemberi perintah. Komponen dasar pada komputer terdiri dari *input*, *process*, *output* dan *storage*. *Input device* terdiri dari *keyboard* dan *mouse*, *Process device* adalah *microprocessor* (ALU, *Internal Communication*, *Registers* dan *control section*), *Output device* terdiri dari monitor dan printer, *Storage external memory* terdiri dari *harddisk*, *Floppy drive*, *CD ROM*, *Magnetic tape*.

Storage internal *memory* terdiri dari RAM dan ROM. Sedangkan komponen Periferal Device merupakan komponen tambahan atau sebagai komponen yang belum ada atau tidak ada sebelumnya. Setelah melakukan perancangan jaringan saraf tiruan untuk mengklasifikasikan penyakit demam berdarah dengan menerapkan metode *backpropagation* maka hasil rancangan seperti dibawah ini:

#### 1. Tampilan Utama

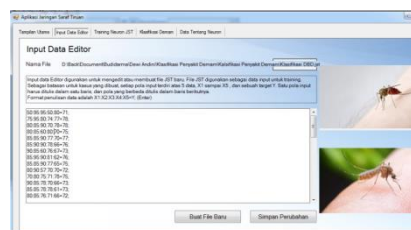
Tampilan utama merupakan interface dari keseluruhan program yang dirancang dimana pada tampilan utama dapat dilihat beberapa menu diantaranya menu Tampilan Utama, Input Data Editor, Training Neuro JST, Klasifikasi Demam dan data Tentang Neuro, maka Bentuk tampilan menu utama dapat dilihat pada gambar dibawah ini.



**Gambar 12.** Tampilan Menu Utama

#### 3. Tampilan Input Data Editor

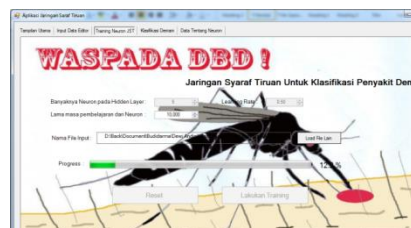
Tampilan menu Input data editor berfungsi untuk menampilkan data yang ada dalam database atau disebut dengan data klasifikasi DBD yang telah dirangkum oleh pihak manajemen rumah Sakit Bakti Medan, sehingga data base tersebut digunakan untuk melakukan training tentang data network neuro pada jaringan saraf tiruan dengan menerapkan metode Backpropagation. Dimana data yang akan diuji terlebih dahulu diproses pada aplikasi yang telah disediakan untuk melakukan training data demi mencapai target yang ditentukan. Tampilan menu editor dapat dilihat pada gambar dibawah ini :



**Gambar 13.** Tampilan Menu Editor

#### 3. Tampilan Training Neuro JST

Tampilan form training neuro JST berfungsi untuk menentukan berapa banyak neuro *Hidden* yang digunakan dan lama masa *training* dilakukan beserta *learning later* untuk menentukan berapa banyak *ranting erro* yang harus dihitung oleh jaringan saraf tiruan dengan menggunakan metode *backpropagation* dalam Neural Networks. Dimana pada form *training neuro JST* memiliki beberapa tombol atau *button* untuk dapat diproses, yakni tombol lakukan training yang berfungsi untuk mentraining data yang ada didalam database dan diolah menjadi hasil neuro untuk menentukan nilai output beserta nilai input dan juga hidden yang telah ditentukan.

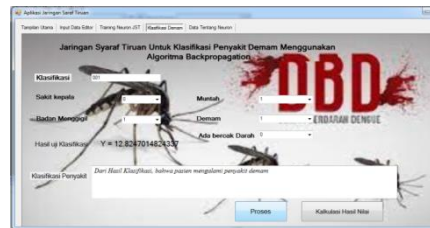


**Gambar 14.** Training Neuro JST

#### 4. Form Klasifikasi Demam

Form klasifikasi berfungsi untuk menampilkan hasil klasifikasi tentang penderita penyakit demam berdarah dengan gejala yang ada, pada form klasifikasi penyakit DBD ini memiliki beberapa *button* yang digunakan untuk memproses data, tombol Kalkulasi hasil nilai berfungsi untuk menentukan hasil klasifikasi  $Y =$  sehingga nilai neuro Networks dapat ditampilkan, Sedangkan tombol proses berfungsi untuk menentukan target apakah pasien yang diklasifikasikan menderita oenyakit demam bardara atau tidak? untuk menentukan pasien terdapat penyakit

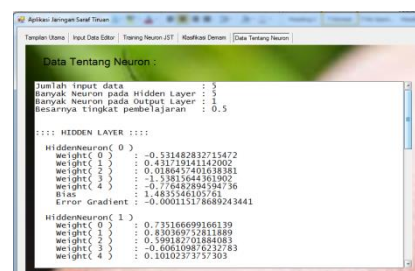
demam berdasarkan kriteria gejala yang dialami. Form klasifikasi penentuan penyakit DBD dapat dilihat pada gambar dibawah ini:



**Gambar 15.** Tampilan Klasifikasi Demam

#### 5. Data Tentang Neuron

Data tentang neuron berfungsi untuk menampilkan hasil akhir dari perhitungan data yang di training, dimana data yang di training menghasilkan jumlah input data sebanyak 5 dan hidden layer sebanyak 5 beserta neuro outputnya ada 1 dan tingkat pembelajarannya error 0,5. Form tentang neuro dapat dilihat pada gambar dibawah ini:



**Gambar 16.** Tampilan Hasil Neuron

Hasil percobaan atau pelatihan jaringan syaraf tiruan menunjukkan bahwa metode backpropagation dalam jaringan saraf tiruan untuk mengklasifikasikan pasien yang mengalami penyakit DBD menghasilkan nilai Output dan Input pada setiap proses pada layer untuk mendapatkan nilai error gradient sesuai dengan yang diharapkan.

## KESIMPULAN

Berdasarkan uraian dan pembahasan sebelumnya, maka dapat diambil kesimpulan bahwa Metode JST *Backpropagation* dapat digunakan untuk melakukan pendeteksian suatu jenis penyakit, gangguan, maupun kasus yang memiliki data masa lalu, dan dengan menggunakan metode *Backpropagation*, Pengujian jaringan saraf dilakukan dengan memasukkan data pelatihan dan pengujian dan diperoleh hasil pengujian sampai 100% sesuai dengan target yang diinginkan. Dan Konfigurasi jaringan terbaik yang diperoleh melalui percobaan (*trial and error*) adalah dengan 5 sel lapisan tersembunyi dan konstanta belajar 0.005 dengan fungsi aktivasi yang digunakan adalah metode *Backpropagation*.

## REFERENCES

- [1] R. L. Riswanto, S. Sutikno, and I. Indriyati, "Aplikasi Prediksi Jumlah Penderita Penyakit Demam Berdarah Dengue Di Kota Semarang Menggunakan Jaringan Syaraf Tiruan Backpropagation," *J. Masy. Inform.*, vol. 5, no. 10, pp. 19–27, 2014, doi: 10.14710/jmasif.5.10.19-27.
- [2] W. Widodo, A. Rachman, and R. Amelia, "Jaringan Syarf Tiruan Prediksi Penyakit Demam Berdarah dengan Menggunakan Metode Backpropagation," *J. IPTEK*, vol. 18, no. 1, pp. 296–304, 2014, [Online]. Available: [http://digilib.batan.go.id/e-prosiding/File Prosiding/Informatika/lkstn/LKSTN2012/Arie-Q2012.pdf](http://digilib.batan.go.id/e-prosiding/File%20Prosiding/Informatika/lkstn/LKSTN2012/Arie-Q2012.pdf).
- [1] R. L. Riswanto, S. Sutikno, and I. Indriyati, "Aplikasi Prediksi Jumlah Penderita Penyakit Demam Berdarah Dengue Di Kota Semarang Menggunakan Jaringan Syaraf Tiruan Backpropagation," *J. Masy. Inform.*, vol. 5, no. 10, pp. 19–27, 2014, doi: 10.14710/jmasif.5.10.19-27.
- [2] W. Widodo, A. Rachman, and R. Amelia, "Jaringan Syarf Tiruan Prediksi Penyakit Demam Berdarah dengan Menggunakan Metode Backpropagation," *J. IPTEK*, vol. 18, no. 1, pp. 296–304, 2014, [Online]. Available: [http://digilib.batan.go.id/e-prosiding/File Prosiding/Informatika/lkstn/LKSTN2012/Arie-Q2012.pdf](http://digilib.batan.go.id/e-prosiding/File%20Prosiding/Informatika/lkstn/LKSTN2012/Arie-Q2012.pdf).
- [3] D. T. Novi Indah Pradasar, F. Trias Pontia W, "Aplikasi Jaringan Syaraf Tiruan Untuk Memprediksi Penyakit Saluran Pernafasan Dengan Metode Backpropagation," *J. Coding Sist. Komput. Univ. Tanjungpura*, vol. 01, no. 1, pp. 20–30, 2013.
- [4] A. Ahyuna and K. Aryasa, "Sistem Pakar Diagnosa Dan Tatalaksana Penyakit Demam Berdarah Dengue Menggunakan Jaringan Saraf Tiruan Backpropagation," *E-JURNAL JUSITI J. Sist. Inf. dan Teknol. Inf.*, vol. 6, no. 1, pp. 1–11, 2018.
- [5] M. D. A. Nasution, J. T. Hardinata, and I. S. Damanik, "Jaringan Syaraf Tiruan Backpropagation untuk Klasifikasi Data Tilang Berdasarkan Jenis Pelanggaran," *Pros. Semin. Nas. Ris. Inf. Sci.*, vol. 1, no. September, p. 547, 2019, doi: 10.30645/senaris.v1i0.61.

- [6] O. K. dan Lazim, Title,” *J. Tunas Bangsa*, pp. 185–197, 2013.
- [7] M. El-Dairi and R. J. House, “Optic nerve hypoplasia,” *Handbook of Pediatric Retinal OCT and the Eye-Brain Connection*. pp. 285–287, 2019, doi: 10.1016/B978-0-323-60984-5.00062-7.