

Penerapan Jaringan Syaraf Tiruan *Backpropagation* Dalam Sistem Pakar Diagnosa Virus TORCH

Application of Backpropagation Artificial Neural Networks in the TORCH Virus Diagnosis Expert System

Erwin Gunawan Walker¹, David²

Teknik Informatika STMIK Pontianak: Jl. Merdeka No. 372, Pontianak 78111

Jurusan Teknik Informatika, STMIK Pontianak

e-mail: erwin_g_walker@yahoo.com, davidliauw@gmail.com

Abstrak

Virus TORCH merupakan singkatan dari virus *Toxoplasma*, *Rubella*, *Cytomegalo Virus*, dan *Herpes*. Virus TORCH menyebabkan banyak anak yang terlahir cacat. Tujuan dari penelitian ini adalah untuk menghasilkan dan membangun sebuah perangkat lunak sistem pakar yang dapat membantu penderita dalam melakukan diagnosa awal terhadap virus TORCH, agar dapat mengurangi resiko anak yang terlahir cacat yang disebabkan oleh virus ini. Bentuk metode penelitian adalah studi kasus dan wawancara dalam mencari tahu tentang virus TORCH. Pembuatan sistem pakar ini menggunakan Visual Basic .NET dalam pembuatan aplikasinya. Sistem pakar yang digunakan adalah forward chaining yang dipadankan dengan jaringan syaraf tiruan *Backpropagation* sebagai pengambilan keputusan. Sistem pakar ini menampilkan pilihan gejala yang dapat dipilih user, dimana setiap pilihan gejala akan membawa user kepada pilihan gejala selanjutnya sampai mendapatkan hasil akhir. Sedangkan database yang digunakan adalah menggunakan *MySQL*. Model perancangan yang digunakan adalah UML (*Unified Modeling Language*) dan *flowchart*. Metode perancangannya adalah *prototype* dan pengujian perangkat lunak menggunakan *black box* untuk menguji sistem yang telah dibangun. Hasil penelitian ini adalah dapat membantu masyarakat agar dapat lebih memahami dan mengetahui virus TORCH yang diderita sehingga dapat segera diobati untuk menghindari anak yang terlahir cacat. Saran kedepannya adalah agar sistem pakar ini dapat bersifat online.

Kata kunci : Sistem Pakar, *Backpropagation*, *Forward Chaining*, Virus TORCH

Abstract

Virus TORCH is an acronym of the virus *Toxoplasma*, *Rubella*, *Cytomegalo virus*, and *Herpes*. TORCH virus causes many children born with disabilities. The purpose of this study is to generate and build an expert system software that can help the patient in making an early diagnosis of the TORCH virus, in order to reduce the risk of children born with disabilities caused by this virus. Method research design is a case study and interviews in finding out about the TORCH virus. This expert system making use of Visual Basic .NET in the making of the application. An expert system is paired with a forward chaining neural network *Backpropagation* as decision-making. This expert system featuring a large selection of symptoms that can be selected user, where each choice of symptoms will bring the user to the selection of the next symptom to get the final result. While the database used is *MySQL*. Design model used is UML (*Unified Modeling Language*) and *flowchart*. Its design method is a *prototype* and testing software using *black box* testing systems have been built. The results of this study is to help the public to better understand and know TORCH suffered virus so that it can be treated promptly to avoid the children who are born with disabilities. Suggestions in the future is that this expert system can be online

Keywords : Expert System, *Backpropagation*, *Forward Chaining*, TORCH Virus

1. PENDAHULUAN

Sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli [4]. Sistem pakar terdiri dari dua bagian yang harus dimiliki yaitu *knowledge base* dan *inference engine*. *Knowledge base* adalah sebuah database yang menyimpan informasi pengetahuan tertentu dan aturan-aturan tentang subjek tertentu, sedangkan *inference engine* adalah bagian dari sistem pakar yang mencoba menggunakan informasi yang diberikan untuk menemukan objek yang sesuai.

Komponen sistem pakar yaitu antar muka pengguna (*user interface*), basis pengetahuan (*knowledge base*), mesin inferensi (*inference machine*), memori kerja (*working memory*), *workplace*, fasilitas penjelasan dan perbaikan pengetahuan. Salah satu implementasi yang diterapkan sistem pakar dalam bidang kesehatan yaitu sistem pakar untuk diagnosa virus TORCH. Virus TORCH adalah singkatan dari *Toxoplasma*, *Rubella*, *Cytomegalo Virus*, dan *Herpes*. Virus TORCH merupakan virus yang dapat menyebabkan anak terlahir cacat bila ibu yang sedang hamil terinfeksi virus ini. Gejala virus TORCH pada awalnya seperti demam, batuk pilek, dan lain-lain. Sehingga pada ibu hamil dapat mewaspadaai pada gejala yang terjadi pada mereka. Agar tidak ada kesalahan diagnosa dan untuk mempermudah masyarakat atau penderita mengetahui sejak dini penyakit yang diderita sehingga tidak terlambat mendapatkan pengobatan. Penggunaan jaringan syaraf tiruan *Backpropagation* digunakan dalam sistem pakar ini.

Penelitian sebelumnya mengusulkan sebuah sistem pakar dengan menggunakan jaringan syaraf tiruan *Backpropagation* untuk mendiagnosa penyakit kulit. Sistem pakar ini mendiagnosa dengan mengambil gambar kulit seseorang yang akan di diagnosa dan kemudian di bandingkan dengan database. Berdasarkan gambar tersebut dapat diketahui apakah penyakit kulit yang diderita orang tersebut [1]. Penelitian lainnya adalah mendiagnosa penyakit gigi. Pengujian dilakukan dengan mengambil 50 sampel data yang belum pernah dilatih hasil menunjukkan bahwa 45 dari 50 (90 %) data yang diuji dapat dikenali oleh sistem [2]. Berikutnya adalah diteksi pengeroposan tulang dengan membandingkan hasil gambar *x-ray*. Sistem kerja di dalam perangkat lunak ini meliputi 3 pemrosesan penting, yaitu proses pengolahan citra dasar, proses reduksi piksel, dan proses jaringan syaraf tiruan dalam mendiagnosa [3].

Penelitian ini menggunakan metode pencarian menggunakan *forward chaining*. *Forward chaining* merupakan teknik pelacakan ke depan yang dimulai dengan informasi yang ada dan penggabungan rule untuk menghasilkan suatu kesimpulan atau tujuan. Sistem ini akan menampilkan pilihan gejala yang dapat dipilih oleh user, dimana setiap pilihan gejala akan membawa user kepada pilihan gejala selanjutnya sampai mendapatkan hasil akhir. Penelitian ini juga menggunakan metode jaringan syaraf tiruan *backpropagation* sebagai penalaran pencarian solusi.

2. METODE PENELITIAN

Penelitian ini berbentuk survey, yaitu mencatat gejala dan solusi virus TORCH dari pakar. Pengumpulan data dilakukan dengan melakukan studi dokumentasi dan observasi. Hasil dari studi dokumentasi dan observasi diagnose virus TORCH dikumpulkan dan menjadi sampel yang digunakan sebagai bahan inferensi jaringan syaraf tiruan. Metode perancangan perangkat lunak menggunakan prototype dengan pendekan UML (*Unified Modeling Language*) dan flowchart.

Untuk metode pengujiannya adalah menggunakan pengujian *black-box* yaitu pengujian memungkinkan perancang perangkat lunak mendapatkan serangkaian input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pembuatan aplikasi

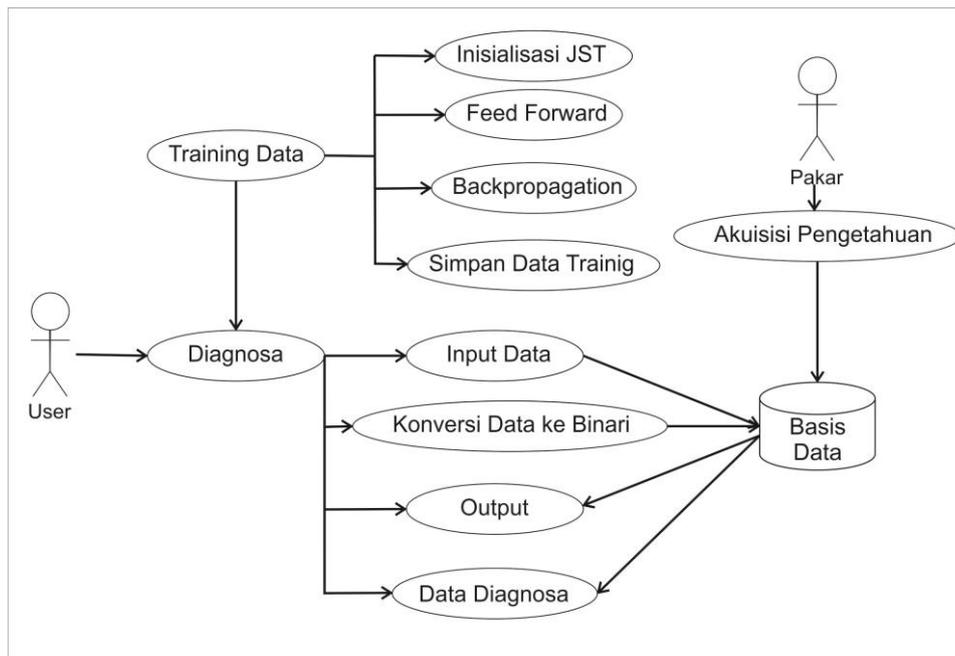
inimenggunakan Visual Basic .net dengan basis data MySQL. Analisis dan perancangan aplikasi dalam penelitian ini menggunakan model UML dan *flowchart*, untuk tahap analisis, metode ini memeriksa kebutuhan yang harus dipenuhi sebuah sistem dan menganalisis data-data yang diperlukan dalam sistem yang dirancang, sedangkan untuk tahap desain, metode ini memfokuskan pada desain pembuatan program perangkat lunak termasuk struktur data, dan arsitektur perangkat lunak [7]. Metode pengujian menggunakan pengujian blackbox yaitu pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan[7].

3. HASIL DAN PEMBAHASAN

Pada penelitian ini, dilakukan indentifikasi kebutuhan yang di perlukan dalam mengembangkan sistem pakar diagnosa virus TORCH. Agar sistem pakar yang di rancang dapat lebih terarah dalam mendiagnosa seorang ibu hamil, maka perlu dilakukan indentifikasi kebutuhan. Idetifikasi kebutuhan fungsional (*functional requirement*) adalah kebutuhan yang berkaitan dengan fungsi atau proses transformasi yang mampu dikerjakan oleh sistem pakar. Dimana sistem pakar diagnosa virus TORCH dapat dapat memberikan beberapa fungsi seperti:

- Sistem pakarmampu menampilkan *formy* yang akan digunakan dalam mendiagnosa seseorang.
- Sistem pakarmampu menampilkan pertanyaan yang akan digunakan dalam mendiagnosa.
- Sistem pakar mampu melakukan inisialisasi struktur dan bobot jaringan saraf tiruan (JST) dan melakukan pelatihan jaringan syaraf tiruan (JST) untuk mengenal pola gejala virus TORCH.
- Sistem pakar mampu memberikansolusi yang didapat dari jawaban *user*, yang telah diproses oleh sistem pakar.

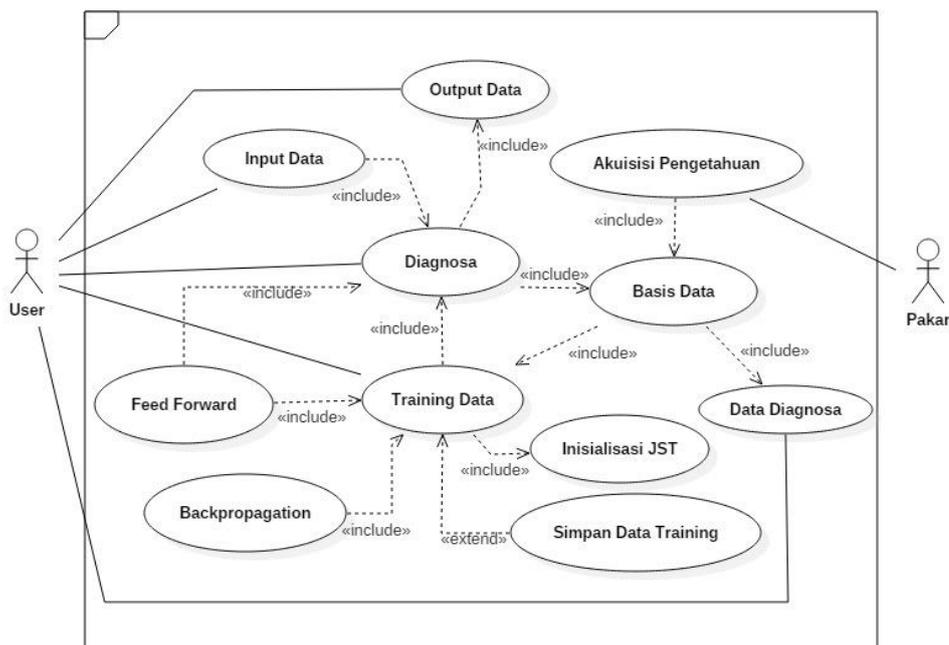
Terdapat beberapa bagian pada arsitektur sistem pakar yang dirancang pada sistem ini diantaranya adalah basis data, basis pengetahuan, akuisisi pengetahuan, mesin inferensi, jaringan syaraf tiruan *Backpropagation*, pengguna dan pakar(gambar 1).



Gambar 1. Arsitektur Sistem Pakar

Dari gambar diatas (Gambar 2) memperlihatkan gambaran umum dari sistem pakar diagnosa virus TORCH. Gambar arsitektur di atas menunjukansistem pakar akanmelakukan *training data*. Di dalam *training data* terdapat beberapa proses, yaitu *backpropagation*, *feedforward*, inisialisasi jaringan syaraf tiruan, dan simpan data *training*. Setelah itu, *user* akan memulai diagnosa. Dimana sistem pakar akan memberikan beberapa pertanyaan. Dari pertanyaan tersebut, *user* akan memberikan *input* yang sesuai dengan pertanyaan dari sistem pakar. Dari *input* yang diberikan *user*, sistem pakar mulai mendiagnosa dan selajutnya data di teruskan ke basis data. Dimana basis data berisikan akuisisi pengetahuan yang telah didapat dari seorang pakar atau dokter kandungan untuk mendiagnosa dan memberikan solusi. Solusi tersebut akan menjadi *output* yang akan ditampilkan kepada *user*. Pada saat diagnosa selesai, maka sistem pakar akan menyimpan data diagnosa dan data tersebut dapat dilihat kembali oleh *user*.

Untuk melihat garis besar dari sistem pakar diagnosa virus TORCH secara fungsional, maka dibuatlah sebuah diagram *use case*. *Use case* didokumentasikan menggunakan *use case diagram* tingkat tinggi, Himpunan *use case* mewakili semua interaksi yang mungkin akan dijelaskan dalam kebutuhan sistem. Aktor dalam proses, yang mungkin manusia atau sistem lainnya direpresentasikan sebagai *stick figure*. Setiap kelas interaksi direpresentasikan sebagai clips yang diberi nama. Garis menghubungkan aktor dengan interaksi. Secara fakultatif, panah dapat ditambahkan ke garis untuk menunjukkan bagaimana interaksi dimulai.

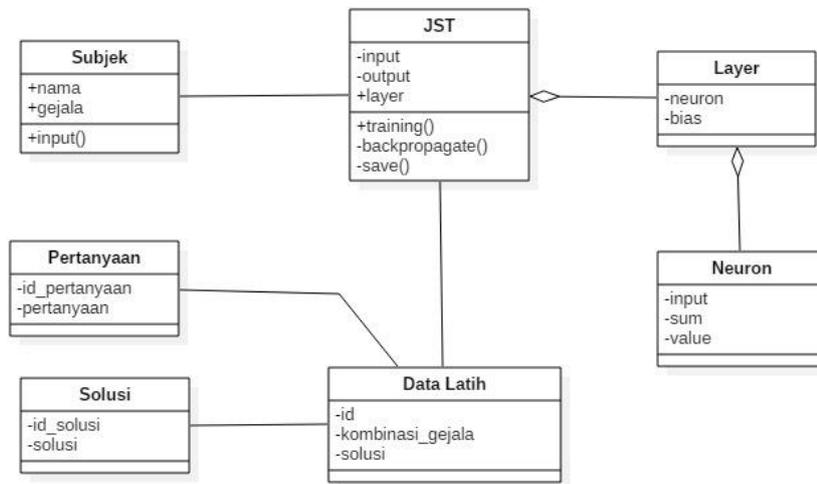


Gambar 2. Diagram *Use case* Sistem Pakar Diagnosa Virus TORCH

Pada gambar diagram *use case* di atas menunjukan cara kerja sistem pakar diagnosa virus TORCH. Saat *user* menggunakan sistem pakar diagnosa virus TORCH, *input* data yang diberikan oleh *user* akan diproses menuju diagnosa, dan pada bagian *training data* akan memproses pola-pola yang pernah terjadi sebelumnya. Setelah itu semua data diteruskan ke basis data untuk memperoleh hasil. Dimana hasil tersebut berasal dari perbandingan dengan data akuisisi pengetahuan yang di dapat dari seorang pakar. Kemudian dari hasil tersebut akan

menjadi *output* yang akan ditamirkan kepada *user*, dan data tersebut akan disimpan di dalam data diagnosa supaya *user* dapat melihat kembalidata yang terdahulu.

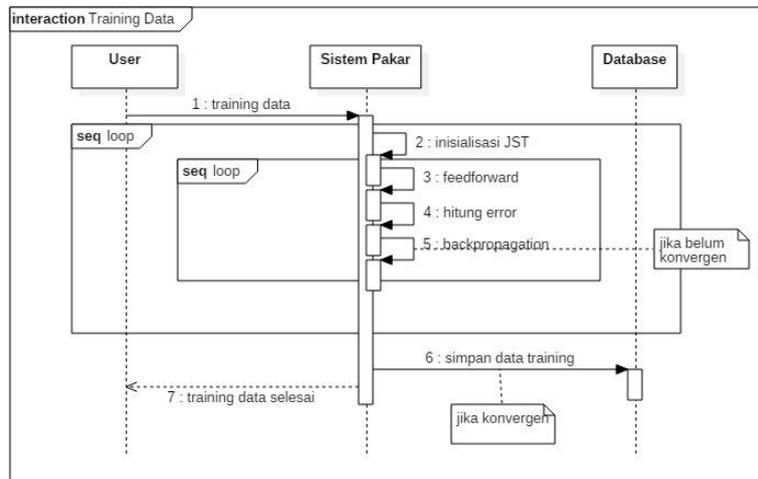
Untuk menggambarkan arsitektur sistem pakar diagnosa virus TORCH, maka dibuatlah *class diagram*. Gambar berikut merupakan *class diagram* dari sistem pakar diagnosa virus TORCH. *Class diagram* dapat dianggap sebagai definisi umum dari satu jenis objek sistem. Sebuah asosiasi adalah hubungan antara kelas-kelas yang menunjukkan hubungan antara kelas-kelas ini. Oleh karena itu, setiap kelas mungkin harus memiliki beberapa pengetahuan dari kelas yang terkait. *Class diagram* dalam UML dapat diekspresikan pada tingkat detail yang berbeda. Ketika mengembangkan model, langkah pertama biasanya untuk melihat dunia, mengidentifikasi objek yang penting dan menyajikannya sebagai sebuah kelas, Cara termudah untuk menulis ini adalah menulis nama kelas dalam sebuah kotak. Dapat juga dengan hanya mencatat keberadaan sebuah asosiasi dengan menggambar garis antara kelas.



Gambar 3. *Class Diagram* Sistem Pakar Diagnosa Virus TORCH

Class diagram diatas menunjukkan gambaran dari sistem pakar yang sedang dikembangkan. Dalam sistem pakar ini, kelas utamanya adalah kelas jaringan syaraf tiruan yang digunakan untuk mendiagnosa. Kelas jaringan syaraf tiruan tersusun dari layer, *input*, dan *output*. layer merupakan kumpulan dari objek neuron. Sedangkan *input* dan *output* berasal dari data latih, dimana data latih berisikan kombinasi gejala yang di dapat dari kelas pertanyaan dan solusi yang di dapat dari kelas solusi.

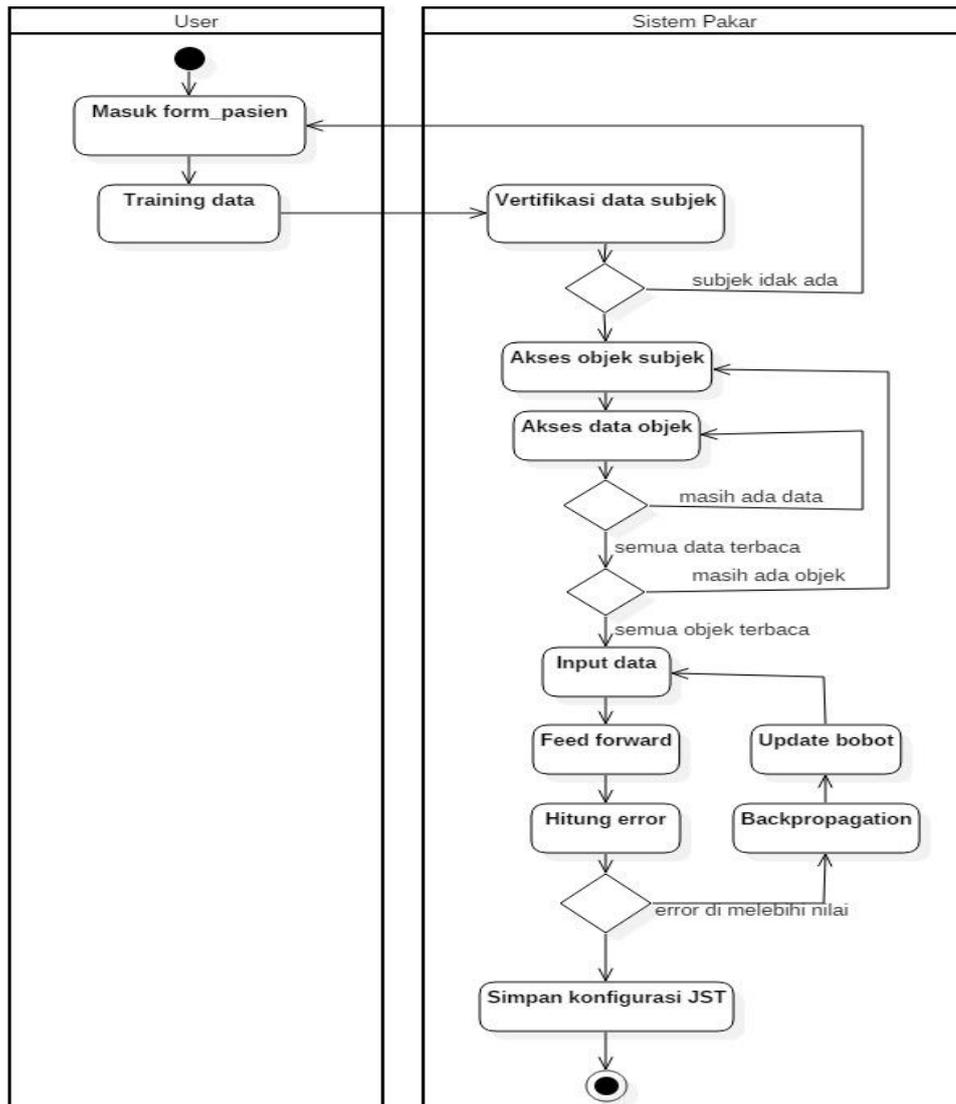
Setelah melakukan *class digram*, diagram berikutnya adalah *sequence diagram*. untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu, Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.



Gambar 4. *Sequence Diagram Trainig Data*

Dari *sequence diagram trainig data* dapat dilihat, *user* akan melakukan *training data* sebelum melakukan diagnosa. Proses *training data* dilakukan dengan melakukan inisialisasi jaringan syaraf tiruan (JST) terlebih dahulu. Proses inisialisasi jaringan syaraf tiruan(JST) dilakukan dengan menginisialisasi beberapa jumlah *layer* dan *neuron* yang telah ditentukan bobot awal secara acak. Data akan dipropagasi maju dalam jaringan, kemudian melakukan perhitungan kesalahan yang terjadi. Jika kesalahan diatas ambang toleransi (belum konvergen) sistem akan mengseksekusi *backpropagation* untuk perbaikan bobot. Jika sudah konvergen, konfigurasi jaringan syaraf tiruan(JST) dan bobot akan di simpan komputer dalam format .NETAK.

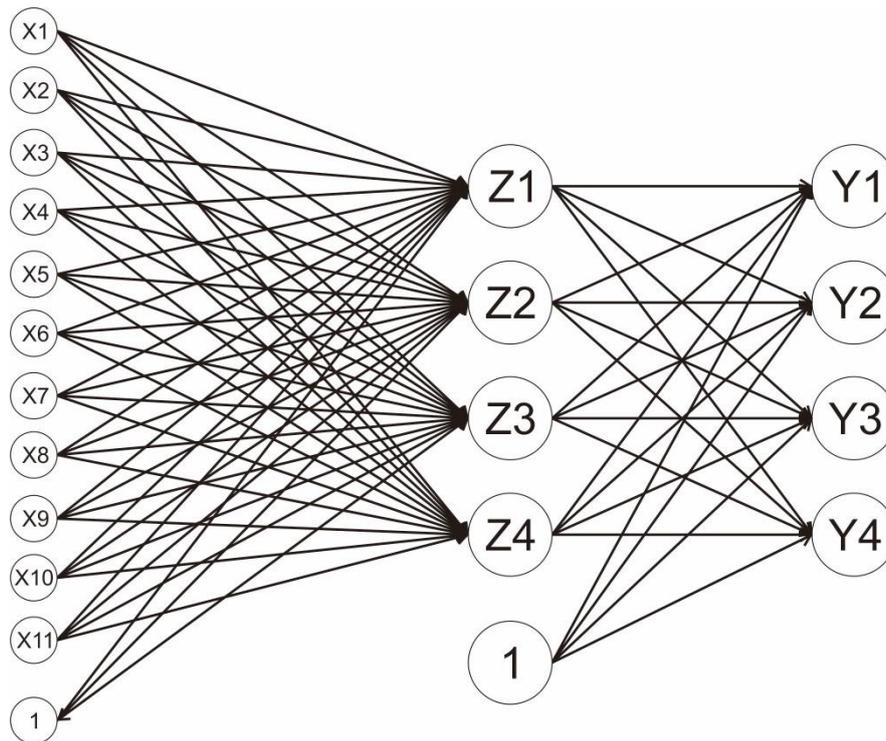
Berikutnya adalah proses dari *activity diagram*. Awal proses ditunjukkan dengan lingkaran tersisi dan di akhir ditunjukkan dengan lingkaran terisi di dalam lingkaran lain. Persegi panjang dengan sudut yang membulat mewakili kegiatan, yaitu sub-proses yang spesifik yang harus dilakukan. UML *activity diagram* , panah mewakili aliran kerja dari satu aktivitas ke aktivitas lainnya. Panah dapat dijelaskan dengan *guards* yang menunjukkan kondisi ketika aliran diambil.



Gambar 5. Activity Diagram Training Data

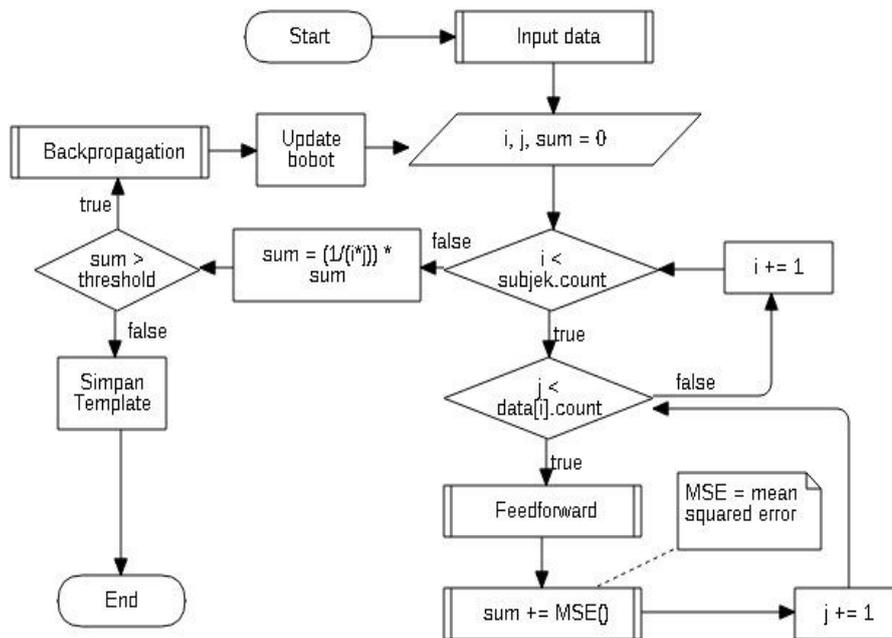
Gambar diatas merupakan *activity diagram* dari *training data*. Saat *user* akan melakukan *training data*, sistem pakar akan memverifikasi keberadaan objek gejala. Setelah itu sistem pakar akan melanjutkan ke dalam jaringan syaraf tiruan (JST) untuk dilatih. Selanjutnya untuk setiap data dan setiap objek akan dimasukkan ke dalam tahap *feedforward*, *backpropagation* dan perbaikan bobot secara berurutan. Proses ini berlangsung sampai tercapainya titik konvergen (JST berhasil dilatih).

Backpropagation adalah salah satu metode dari jaringan syaraf tiruan yang dapat diaplikasikan dengan baik dalam bidang peramalan (*forecasting*). *Backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan mengenali pola yang digunakan selama *training* serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa namun tidak sama dengan pola yang dipakai selama pelatihan. Untuk arsitektur jaringan syaraf tiruan yang digunakan adalah arsitektur jaringan syaraf tiruan *backpropagation*. Arsitektur tersebut dapat dilihat pada gambar dibawah ini (Gambar 2) [8].



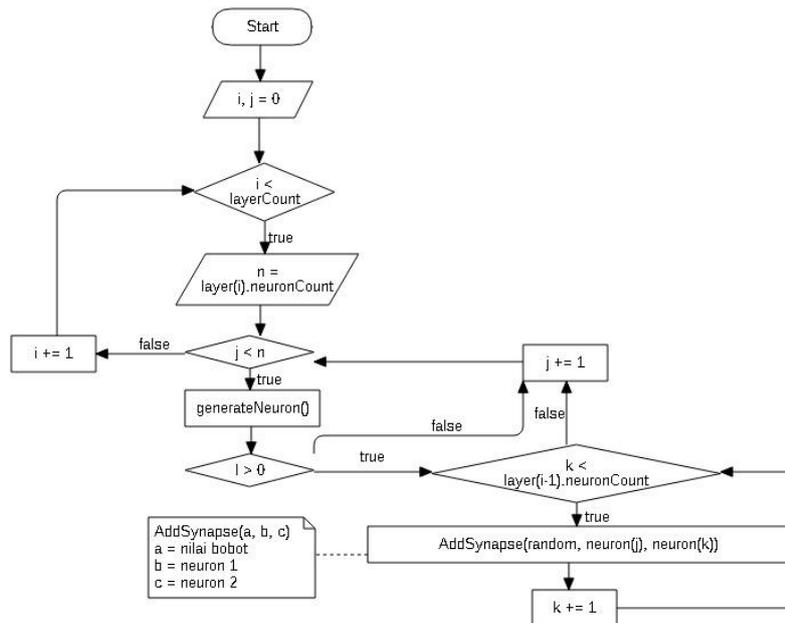
Gambar 6. Arsitektur Jaringan Syaraf Tiruan *Backpropagation*

Gambar diatas menunjukkan X adalah *input layer*, Z adalah *hidden layer*, dan Y adalah *output layer*. Saat sistem pakar menerima data masukan, maka data tersebut akan di masukan kedalam bagian X. Kemudian data tersebut akan dihitung dan dilanjutkan ke *hidden layer*. Setelah dihitung, maka data tersebut akan masuk ke *output layer* dan kemudian ditampilkan. Setelah itu, proses *backpropagation* akan dilakukan. Dimana proses *backpropagation* adalah proses menghitung kesalahan dari setiap *layer*. Sehingga proses *backpropagation* akan di mulai dari *output layer*, kemudian *hidden layer*, dan terakhir *input layer*. Setelah proses *backpropagation* selesai, maka sistem pakar akan merubah bobot tersebut sesuai perhitungan kesalahan *backpropagation*.



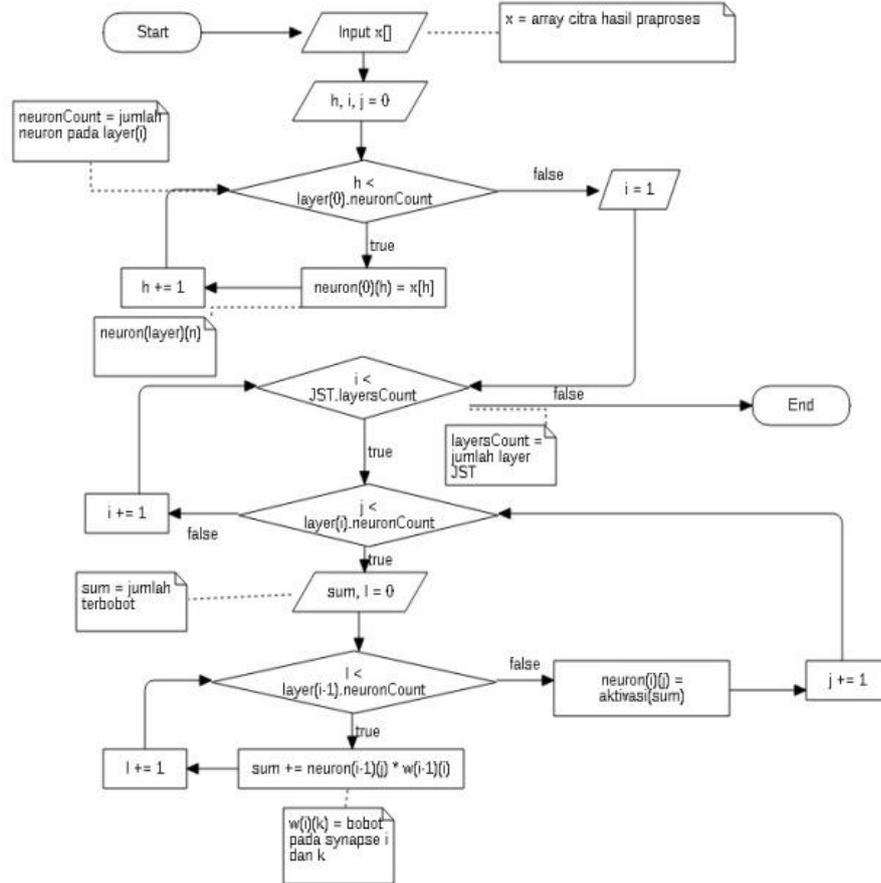
Gambar 7. Flowchart Training Data

Pada gambar *flowcharttrainin* (Gambar 3) data dimulai dengan memasukkan pola data gejala. Setelah pola data gejala diterima, akan dilakukan propagasi maju dan perhitungan kesalahan. Kemudian dilakukan pemeriksaan nilai kesalahan dari jaringan yang akan dilatih. Jika nilai kesalahan diatas nilai yang telah ditentukan maka akan dilanjutkan ke tahap propagasi mundur. Perhitungan kesalahan yang di lakukan berasal dari masing – masing sinapsis yang menghubungkan antar *neuron* pada jaringa syaraf tiruan (JST) yang dilatih. Perhitungan ini akan menghasilkan nilai perubahan bobot yang perlu dilakukan pada setiap bobot yang ada dan dapat dilanjutkan dengan proses perbaruan nilai bobot. Jika nilai kesalahan berada dibawah nilai maka konfigurasi jaringan akan disimpan.



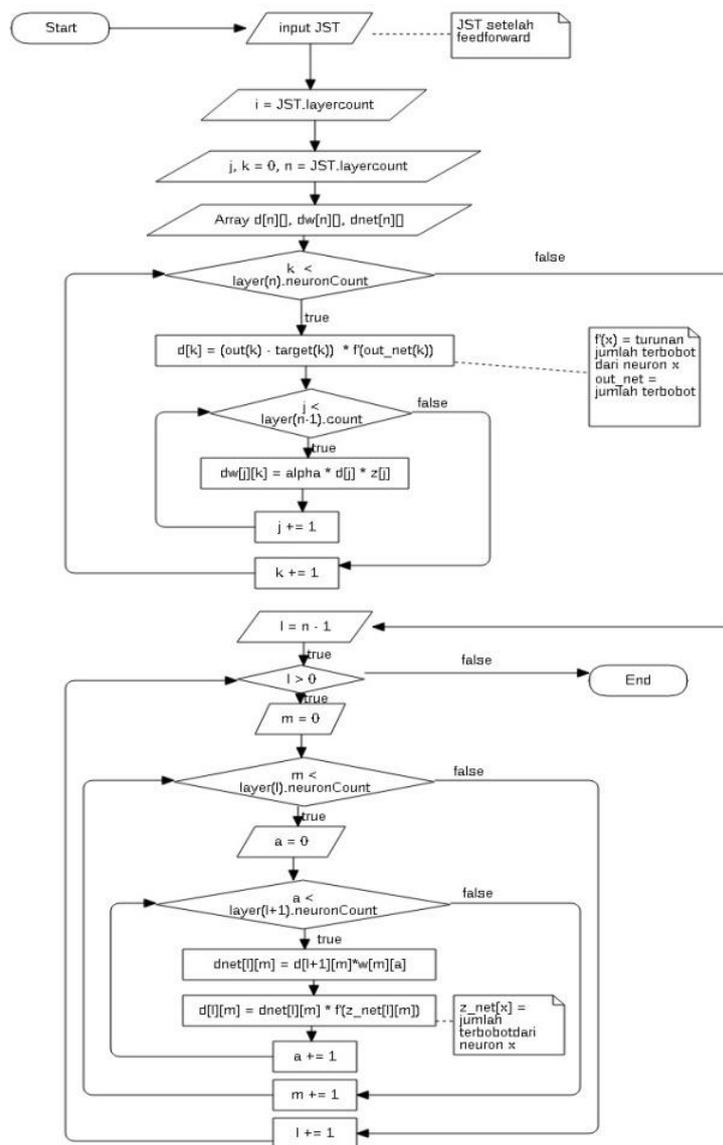
Gambar 8. *Flowchart* Inisialisasi JST

Pada gambar diatas merupakan *flowchart* inialisasi jaringan syaraf tiruan (JST) (Gambar 4). Dalam inialisasi jaringan syaraf tiruan (JST), sistem pakar akan membangun tiga *layer*, yaitu *input*, *hidden*, dan *output*. Pertama setiap neuron akan di mulai dari *hidden layer* yang akan dihubungkan dengan masing-masing *neuron* pada *layer* sebelumnya. *Neuron* tersebut dihubungkan dengan sinapsis yang disertakan dengan nilai bobot awal yang bersifat acak.



Gambar 5. *Flowchart Feedforward*

Gambar diatas merupakan *flowchart feedforward*. *Feedforward* (Gambar 5) merupakan tahap propagasi maju, dimana setiap *neuron* masukan akan dilakukan penjumlahan terbobot terhadap masing – masing bobot pada setiap sinapsis yang menghubungkan *neuron* yang bersangkutan ke *layer* selanjutnya. Kemudian dari penjumlahan bobot tersebut akan di masukan ke dalam sebuah fungsi aktivasi yang akan menjadi nilai input untuk *layer* selanjutnya. Proses ini dilakukan untuk setiap *layer* dan setiap *neuron* pada jaringan syaraf tiruan (JST) yang sudah dibentuk.



Gambar 9. Flowchart Backpropagation

Gambar diatas merupakan *flowchart backpropagation* (Gambar 6). *Backpropagation* merupakan tahap propagasi mundur. *Backpropagation* dimulai dengan perhitungan kesalahan pada *layer* terakhir atau *layer output*. Setelah itu akan dicari tanggung jawab dari masing-masing sinapsis yang terhubung atas kesalahan yang terjadi dengan menghitung perubahan bobot yang ada. Proses ini dilakukan secara iteratif terhadap setiap *neuron* pada masing – masing *layer* dari *layer output* ke *layer input*. Hasil tersebut akan digunakan untuk perbaruan nilai bobot pada tahap selanjutnya.

Secara rinci algoritma pelatihan jaringan backpropagation dapat diuraikan sebagai berikut [9]:

- Langkah 0 : Inisialisasi bobot (setiap bobot diberi nilai acak antara 0 - 1).
- Langkah 1 : Selama kondisi berhenti belum terpenuhi lakukan langkah 2-3.
- Langkah 2 : Lakukan Langkah 3-13 sebanyak jumlah pelatihan yang diinginkan.
- Langkah 3 : Untuk *output layer* dan setiap *hidden layer* lakukan langkah 4-13.

- Langkah 4 : Hitung *input* setiap *node* pada *hidden layer*
 Langkah 5 : Hitung *output* setiap *node* pada *hidden layer* dengan fungsi aktivasi.
 Langkah 6 : Hitung *input* setiap *node* pada *output layer*.
 Langkah 7 : Hitung *output* pada setiap *node* pada *output layer*.
 Langkah 8 : Hitung *error* setiap *node* pada *output layer* dengan fungsi deaktivasi.
 Langkah 9 : Hitung perubahan bobot pada setiap *node* pada setiap *hidden layer*.
 Langkah 10 : Hitung *error* setiap *node* pada *output layer* dengan fungsi deaktivasi.
 Langkah 11 : Hitung perubahan bobot pada setiap *node* pada setiap *hidden layer*.
 Langkah 12 : *Update* bobot pada setiap *node* pada *output layer*.
 Langkah 13 : *Update* bobot pada setiap *node* pada setiap *hidden layer*.

Setelah pelatihan, jaringan saraf *backpropagation* diaplikasikan dengan hanya menggunakan tahap perambatan maju dari algoritma pelatihan. Prosedur aplikasinya adalah sebagai berikut (Kusrini, 2008):

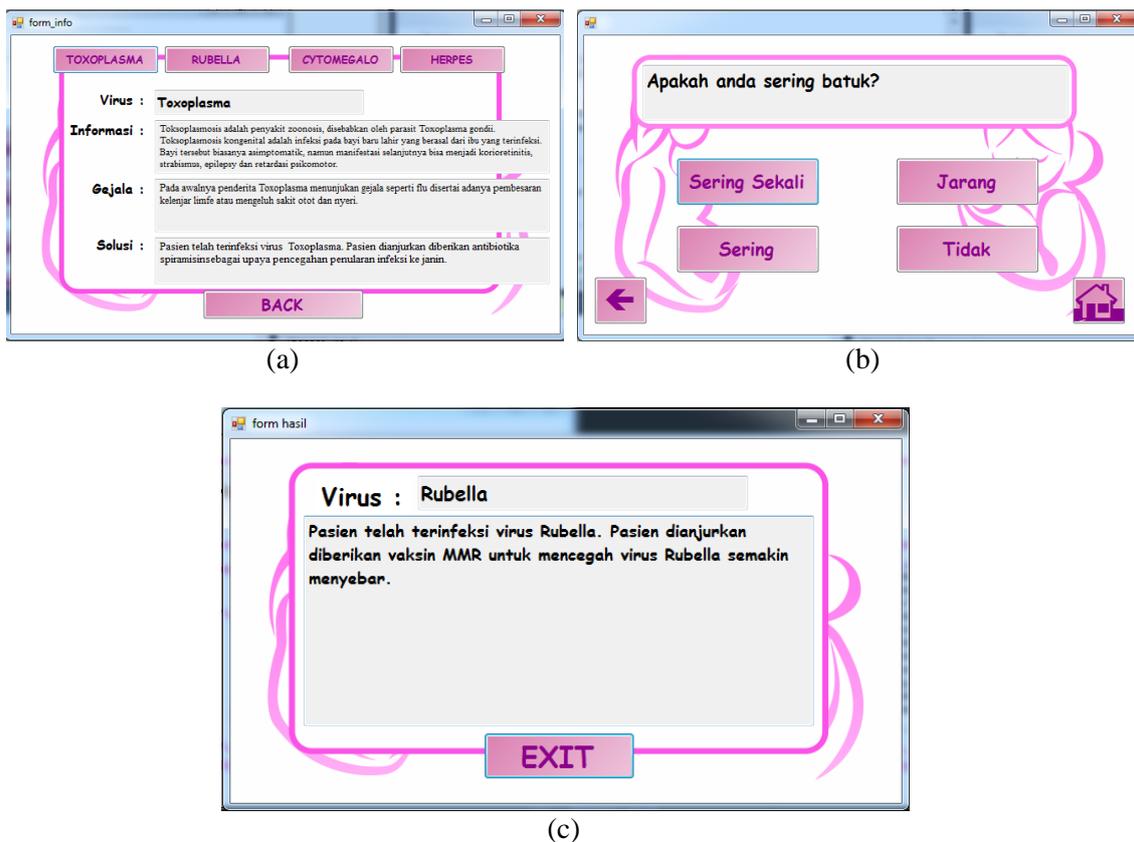
- Langkah 0 : Inisialisasi bobot (dari algoritma pelatihan).
 Langkah 1 : Untuk tiap vector masukan, lakukan langkah 2-4.
 Langkah 2 : for $i = 1, \dots,$
 Langkah 3 : for $j = 1, \dots, p :$
 Langkah 4 : for $k = 1, \dots, m :$
 Langkah 5: Jika $y_k \geq 0,5$ maka $y_k = 1$, else $y_k = 0$..

Dari studi kasus dan observasi di dapatkan kesimpulan sebagai berikut (Tabel 1):

Tabel 1. Tabel Pola Gejala Virus TORCH

	Toxoplasma	Rubella	Cyto Megalo Virus	Herpes
Batuk	-	✓	-	-
Demam	✓	✓	✓	-
Flu	✓	✓	-	-
Nyeri	✓	-	-	-
Mata Gatal	-	✓	-	-
Iritasi Kulit	-	✓	-	-
Tenggorokan Sakit	✓	✓	-	-
Lemah	✓	-	✓	-
Lambung Nyeri	-	-	✓	-
Pembengkakan	-	-	✓	-
Cacar Aie	-	-	-	✓

Supaya sistem pakar diagnosa virus TORCH dapat mudah digunakan, maka tampilan sistem pakar ini dibuat sesederhana mungkin. *User interface* adalah bagian dari perangkat lunak yang dapat dirasakan oleh panca indra pengguna, baik penglihatan maupun indra yang lain.



Gambar 7. Tampilan Halaman (a) form_info;(b) form_diagnosa;(c) form_hasil

Dari gambar di atas dapat di lihat, hasil aplikasi yang dibangun akan tampil *user-friendly* agar mudah digunakan user. Penerapan sistem pakar dalam mendiagnosa virus TORCH ini dapat membantu penderita dalam mengetahui awal virus TORCH untuk dapat mengetahui bahaya dari TORCH, agar dapat mencegah atau mengurangi resiko anak yang terlahir cacat yang di timbulkan akibat terlambatnya penanganan virus TORCH. Selain dapat melakukan diagnosa dan memberikan nilai kepercayaan serta solusi yang berhubungan dengan virus TORCH, sistem ini juga dirancang untuk dapat menampilkan informasi mengenai virus TORCH serta dapat melakukan penambahan dan perubahan data mengenai virus TORCH. Penerapan metode forward chaining dan jaringan syaraf tiruan *Backpropagation* pada aplikasi diagnosa virus TORCH ini dapat mempermudah dan memberikan penyelesaian seberapa pasti user menderita virus TORCH.

untuk pengujian dengan kasus, terdapat 3 ibu hamil yang dijadikan sebagai contoh kasus. Ibu yang pertama adalah ibu Yuli. Ibu Yuli memberitahukan bahwa dia merasa panas demam disertai pilek, nyeri di bagian tangan, sakit tenggorokan, dan mulai merasa lemah. ibu kedua yaitu ibu Desi mengeluhkan bahwa dia merasa demam disertai batuk pilek, matanya mulai gatal tetapi tidak ada debu yang masuk ke mata, kemudian kulitnya mulai gatal seperti terjadi iritasi kulit dan adanya pembengkakan, dan sering sakit di bagian perut di sertai badan yang mulai lemah. Ibu yang terakhir adalah ibu Yeli. Ibu yeli menatakan bahwa di seperti terserang cacar air pada saat dia hamil. Dari data diatas, dokter mengatakan bahwa ibu Yuli hanya sakit biasa, sedangkan ibu Desi terinfeksi virus *rubella*, dan ibu Yeli terinfeksi virus Herpes. Data diatas dapat dilihat pada tabel dibawah (Tabel 2).

Tabel 2. Tabel Data Gejala

Nama	Pola Gejala											Virus
	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	
Yuli	0	1	1	1	0	0	1	1	0	0	0	Tidak Sakit
Desi	1	1	1	0	1	1	1	0	0	0	0	<i>Rubella</i>
Yeli	0	0	0	0	0	0	0	0	0	0	1	<i>Herpes</i>

Data di atas adalah data berdasarkan berdasarkan diagnosa dari dokter. setelah data tersebut di ujikan kepada sistem pakar, hasil yang keluar hampir sama dengan diagnosa dokter. data hasil uji sistem pakar dapat dilihat pada tabel di bawah (tabel 3).

Tabel 3. Tabel Data Gejala *User* Setelah Diproses

Nama	Pola Gejala											Virus
	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	
Yuli	3,6%	7,5%	8,2%	9,3%	1,2%	4,3%	7,9%	8,1%	1,5%	3,2%	4,3%	<i>Toxoplasma</i>
Desi	8,3%	9,4%	7,9%	5,1%	7,5%	8,2%	7,2%	3,1%	2,1%	1,6%	2,5%	<i>Rubella</i>
Yeli	2,6%	0,7%	3,5%	1,5%	2,2%	3,5%	2,1%	3,1%	2,2%	4,1%	9,4%	<i>Herpes</i>

Tabel diatas adalah hasil dari proses diagnosa setelah jawaban user diproses. Pola gejala diambil dari nilai dari paling besar dan telah dibandingkan dengan data di database. Setelah itu, data akan ditampilkan. Dari hasil di atas di dapatkan, bahwa terjadi kesalahan pada ibu Yuli yang tidak sakit tetapi dikatakan sakit oleh sistem pakar. Sedangkan ibu yang lain mendapatkan hasil yang sama dengan diagnosa dokter.

4. KESIMPULAN

Hasil sistem pakar diagnosa virus TORCH ini menghasilkan kesimpulan sebagai berikut:

- Metode jaringan syaraf tiruan *backpropagation* dapat diimplementasikan pada sistem pakar yang dibangun menggunakan visual basic.net.
- Sistem pakar ini mampu mendiagnosa virus TORCH berdasarkan gejala-gejalanya.

Sistem pakar ini dapat digunakan *user* untuk melakukan diagnosa virus TORCH tanpa harus bertemu dengan seorang pakar.

5. SARAN

Ada beberapa saran yang dapat diberikan berkaitan dengan pengembangan asistem pakar ini. Saran untuk penelitian yang ingin melanjutkan pengembangan sistem pakar ini adalah:

- Pengembangan sistem pakar ini di kemudian hari adalah dapat membuat pilihan untuk menambahkan gejala.
- Sistem pakar ini berjalan secara offline, sehingga diharapkan penelitian berikutnya dapat membuat sistem pakar ini berjalan secara *online*.

Dapat mengembangkan sistem pakar ini dengan menggunakan metode lain yang lebih efektif dan efisien.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada orang tua dan keluarga yang telah memberikan dukungan baik moral maupun materil kepada penulis untuk selalu berusaha mencapai hasil yang baik. Dan rekan-rekan seperjuangan khususnya di Teknik Informatika angkatan 2013/2014 konsentrasi kecerdasan buatan yang telah memberikan banyak bantuan, dukungan, saran, dan nasehat yang sangat berarti bagi penulis.

DAFTAR PUSTAKA

- [1] Indrawaty, A. Y., Hermana, A. N., & Ramadhan, A. (2012). Implementasi Model Backpropagation Dalam Mengenali Pola Gambar Untuk Mendiagnose Penyakit Kulit. *J. Inform*, 3(1), 1-7.
- [2] Aryasa, K. (2012). Expert System Diagnosa Jenis Penyakit Gigi Menggunakan Jaringan Saraf Tiruan Backpropagation. *CSRID*, 4.
- [3] David, D., & Kosasi, S. (2013). Penerapan Algoritma Jaringan Saraf Tiruan Backpropagation Untuk Pengenalan Pola Tandatangan. *Jurnal Teknologi*, 6(2), 139-146.
- [4] Mardianto, I., & Pratiwi, D. (2008). Sistem deteksi penyakit pengeroposan tulang dengan metode jaringan syaraf tiruan backpropagation dan representasi ciri dalam ruang eigen. *CommIT (Communication and Information Technology) Journal*, 2(1), 69-80.
- [5] David, D. (2015). Pengenalan Pola Plat Nomor Kendaraan Menggunakan Jaringan Syaraf Tiruan Backpropagation. *Sisfotenika*, 3(1), 71-80.
- [6] David, D. (2017). Penerapan Jaringan Syaraf Tiruan Perceptron Dalam Sistem Pakar Diagnosa Penyakit Pada Kambing. *E-Proceedings KNS&I STIKOM Bali*, 85-90.