

Identifikasi Penyakit Diabetes Millitus Menggunakan Jaringan Syaraf Tiruan Dengan Metode Perambatan-Balik (Backpropagation)

Sriyanto dan Sutedi

Fakultas Ilmu Komputer, Informatics & Business Institute Darmajaya

Jl. Z.A Pagar Alam No 93, Bandar Lampung - Indonesia 35142

Telp. (0721) 787214 Fax. (0721)700261

e-mail : sriyantommsi@yahoo.com / b4nk_t3d1@yahoo.com

ABSTRACT

Diabetes Melitus (DM) is dangerous disease that affect many of the various layer of work society. This disease is not easy to accurately recognized by the general society. So we need to develop a system that can identify accurately. System is built using neural networks with backpropagation methods and the function activation sigmoid. Neural network architecture using 8 input layer, 2 output layer and 5 hidden layer. The results show that this methods succesfully clasifies data diabetics and non diabetics with near 100% accuracy rate.

Keyword : Neural Network, Backpropagation, diabets

ABSTRAK

Diabetes Melitus (DM) adalah penyakit berbahaya yang mempengaruhi banyak dari berbagai lapisan masyarakat kerja. Penyakit ini tidak mudah untuk secara akurat diakui oleh masyarakat umum. Jadi kita perlu mengembangkan sistem yang dapat mengidentifikasi secara akurat. Sistem dibangun dengan menggunakan jaringan syaraf tiruan dengan metode backpropagation dan sigmoid aktivasi fungsi. Arsitektur jaringan saraf menggunakan 8 input layer, 2 layer output dan 5 lapisan tersembunyi. Hasil penelitian menunjukkan bahwa metode ini berhasil clasifies penderita diabetes data dan non penderita diabetes dengan dekat tingkat akurasi 100%.

Kata kunci: Jaringan Syaraf, Backpropagation, diabets

1. PENDAHULUAN

Diabetes adalah salah satu penyakit yang paling sering diderita dan penyakit kronik yang serius di Indonesia saat ini. *Diabetes Mellitus* (DM) yang umumnya dikenal sebagai kencing manis adalah penyakit yang ditandai dengan hiperglisemia (peningkatan kadar gula darah) yang terus-menerus dan bervariasi, terutama setelah makan. DM adalah keadaan *hiperglisemia* kronik disertai berbagai kelainan metabolik akibat gangguan hormonal, yang menimbulkan berbagai komplikasi kronik pada mata, ginjal, dan pembuluh darah, disertai lesi pada *membran basalis* dalam pemeriksaan dengan mikroskop electron. Semua jenis DM memiliki gejala yang mirip dan komplikasi pada tingkat lanjut. *Hiperglisemia* sendiri dapat menyebabkan *dehidrasi* dan *ketoasidosis*. Komplikasi jangka lama termasuk penyakit *kardiovaskular* (risiko ganda), kegagalan kronis ginjal (penyebab utama dialisis), kerusakan retina yang dapat menyebabkan kebutaan, serta kerusakan saraf yang

dapat menyebabkan impotensi dan gangren dengan risiko amputasi. Komplikasi yang lebih serius lebih umum bila kontrol kadar gula darah buruk. Organisasi Kesehatan Dunia (WHO) mengakui tiga bentuk *Diabetes Mellitus*, yaitu : (1) *DM tipe 1* penyebab diabetes yang utama adalah karena kurangnya produksi insulin; (2) *DM tipe 2*, penyebab kurang sensitifnya jaringan tubuh terhadap insulin ; (3) *DM pada kehamilan*, penyebab resistensi insulin pada masa kehamilan.

DM tipe 1 membutuhkan terapi insulin, sedangkan DM tipe 2 hanya membutuhkan insulin apabila penanganan sebelumnya tidak efektif. *Diabetes Mellitus* pada kehamilan umumnya sembuh dengan sendirinya setelah persalinan. Pemahaman dan partisipasi pasien sangat penting karena tingkat glukosa darah berubah terus. Kesuksesan menjaga kadar gula darah dalam batasan normal dapat mencegah terjadinya komplikasi. Faktor lainnya yang dapat mengurangi komplikasi adalah: berhenti merokok,

mengoptimalkan kadar kolesterol, menjaga berat tubuh yang stabil, mengontrol tekanan darah tinggi, dan melakukan olah raga teratur.

Penyakit DM dapat diderita oleh berbagai lapisan umur, dari mulai anak-anak sampai dengan orang lanjut usia. Untuk dapat mengenali apakah seseorang menderita penyakit *Diabetes Millitus* (DM) dapat dilihat dari gejala-gejala maupun dengan cara pemeriksaan darah. Tentu saja hal ini tidak mudah untuk dilakukan dan memerlukan bantuan dokter atau tenaga kesehatan lainnya untuk dapat mengidentifikasi penyakit ini dengan akurat.

Oleh sebab itu, diperlukan sebuah sistem yang mampu menganalisa karakteristik *Diabetes Millitus* sehingga mempermudah dalam mengidentifikasi seseorang terkena *Diabetes Millitus* dan terkena jenis yang mana. Dengan menggunakan jaringan saraf tiruan metode perambatan-balik (*backpropagation*) dapat dibuat sebuah sistem yang mampu menganalisis dan

mengidentifikasi penyakit *Diabetes Millitus* seseorang. Jaringan saraf tiruan akan melakukan diagnosa, dengan menyelidiki nilai variabel biner dari data, apakah pasien menunjukkan tanda-tanda diabetes menurut kriteria yang telah ditentukan oleh Organisasi Kesehatan Dunia (WHO) yaitu jika ditemukan konsentrasi glukosa plasma 2 jam dalam tes glukosa oral sedikitnya 200 mg / dl atau jika ditemukan dalam perawatan medis yang rutin (berkala).

Jaringan saraf tiruan dibangun menggunakan algoritma *Feed-Forward Backpropagation*. Langkah awal yang dilakukan adalah dengan melakukan training (pelatihan) dengan beberapa data. Setelah pelatihan, jaringan saraf tiruan diuji. Diharapkan hasil pengujian dengan jaringan saraf tiruan yang dirancang dapat memberikan perkiraan dengan keakuratan lebih dari 90% dengan memperhatikan beberapa faktor pengujian seperti jumlah berapakah hamil, konsentrasi glukosa plasma 2 jam dalam tes toleransi glukosa oral, tekanan darah diastolik (mm Hg), Triceps ketebalan lipatan (mm), 2-Jam

serum insulin ($\mu\text{U} / \text{ml}$) , Indeks Massa Tubuh (berat dalam $\text{kg} / (\text{tinggi dalam m})^2$), silsilah diabetes, Usia (tahun).

Jaringan syaraf tiruan (*Artificial Neural Network*) merupakan salah satu metode *soft computing* yang perkembangannya diinspirasi dari cara kerja syaraf manusia. Implementasi dan riset jaringan syaraf tiruan berjalan pesat di berbagai bidang seperti pemetaan pola dan klasifikasi pola, analisis citra dan pengkodean, pengolahan sinyal, optimasi, manipulasi grafis, pengenalan karakter, robotik, pengolahan pengetahuan dan sistem pakar, diagnosis kesehatan, peredaman kebisingan, dan sebagainya (Marimin 2009).

Di bidang Biometrik, jaringan syaraf tiruan berperan dalam pengklasifikasian pola bersama dengan metode *soft computing* lainnya seperti *fuzzy logic* dan algoritma genetika hingga bisa diimplementasikan secara nyata saat ini.

Metode algoritma jaringan syaraf tiruan banyak variasinya dari yang

paling sederhana seperti *Perceptron*, kemudian berkembang menjadi *Adaline* (*Adaptive Linear*) dan *Madaline* (*Multi-Adaline*) di bidang pengolahan sinyal. Metode paling terkenal hingga saat ini adalah algoritma Propagasi balik (*Backpropagation*) yang dikembangkan secara formal oleh Werbos, dan kemudian Parker serta Rummelhart dan McClelland (Siang 2009).

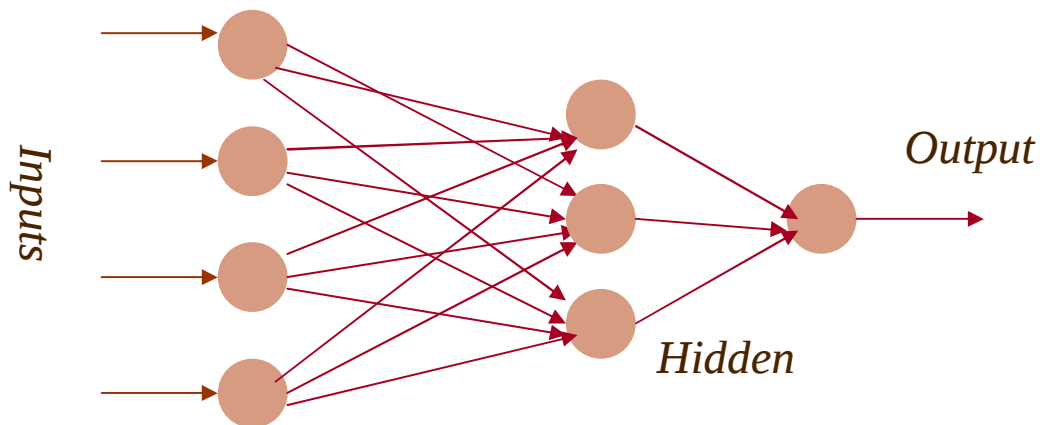
Jaringan syaraf tiruan Propagasi balik (*Backpropagation*) menggunakan pelatihan terarah (*supervised training*) yang memudahkan dalam melakukan pembelajaran. Dengan merancang data pelatihan yang berupa pasangan pola input dan target output yang diinginkan, maka jaringan syaraf tiruan Propagasi balik (*Backpropagation*) akan melakukan pembelajaran sedemikian rupa sehingga diperoleh hasil yang paling mendekati data pelatihan.

Jaringan syaraf tiruan bersifat lebih adaptif dan fleksibel dibandingkan dengan metode *soft computing* lainnya sehingga

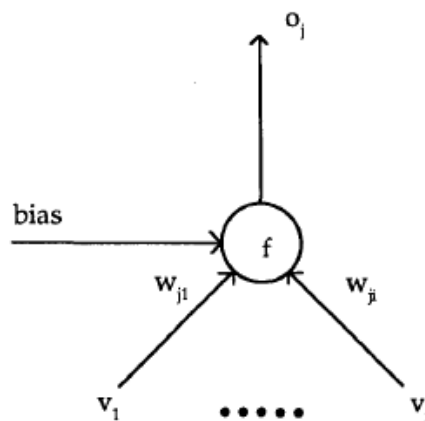
memudahkan dalam melakukan ragam variasi dan kombinasi percobaan tanpa memerlukan banyak perubahan pada kode program.

Arsitektur jaringan syaraf tiruan propagasi balik (*Backpropagation*) cukup sederhana dan terdiri atas

lapisan input dan lapisan output serta di antaranya terdapat satu atau lebih lapisan antara yang disebut *Hidden Layer*. Di setiap lapisan terdapat unit-unit sel yang saling berhubungan satu sama lain membentuk jaringan seperti gambar 1. dibawah ini.



Gambar 1 Arsitektur Lapisan Jaringan Syaraf Tiruan (JST)



Gambar 2 Pemrosesan Di Satu Sel Neuron

Gambar 2 menunjukkan koneksi antar sel mempunyai bobot W . Output O adalah fungsi f dari perkalian bobot W dengan input V yang merupakan output dari sel *neuron* sebelumnya yang dinyatakan sebagai berikut.

$$o_j = f\left(\sum_i w_{ji} v_i\right)$$

Bias diikutsertakan dalam penjumlahan dan diasumsikan bernilai 1. Namun, *bias* ini bersifat pilihan atau bisa digunakan bisa tidak (Siang 2009).

f adalah fungsi aktivasi dan yang sering digunakan adalah fungsi sigmoid :

$$o_j = \frac{1}{1 + e^{-(\sum_i w_{ji} v_i)}}$$

Bobot W akan menentukan besarnya nilai output dan nilai bobot W selalu berubah selama tahapan pembelajaran. Awalnya bobot diberi inisialisasi nilai acak pada rentang nilai tertentu. Akhirnya setelah pembelajaran selesai, bobot W mencapai nilai yang sedemikian rupa, sehingga untuk setiap input yang diberikan maka output bernilai sesuai dengan pola

pembelajaran yang telah dilakukan. Bobot W adalah pengetahuan yang disimpan dalam JST sebagaimana halnya informasi yang disimpan dalam jaringan syaraf otak.

Jumlah sel neuron di *hidden layer* akan menentukan jumlah bobot W yang dapat disimpan. Beberapa literatur JST mengatakan bahwa : (1) Jumlah sel neuron N_H terlalu besar mengakibatkan *training error* menjadi kecil, namun *test error* menjadi tinggi; (2) Jumlah sel neuron N_H terlalu sedikit berakibat *training error* menjadi tinggi dan *test error* menjadi tinggi.

Dengan mencari nilai optimum, yakni jumlah sel neuron N_H tidak terlalu besar atau terlalu kecil sedemikian rupa sehingga *training error* dan *test error* mencapai nilai optimum yang cukup memadai.

JST diarahkan dalam proses pembelajaran dengan nilai yang diinginkan (target) sedemikian rupa sehingga JST akan melakukan perubahan bobot W dan nilai output

akan mendekati nilai target. Nilai kesalahan target dengan output dinyatakan dalam persamaan berikut.

$$\delta_{pi} = (t_{pi} - o_{pi}) f'_j(\text{net}_{pi})$$

Nilai kesalahan pada *hidden unit* akan ditentukan secara berulang (rekursif) sedemikian rupa oleh unit dan bobot lain yang saling berhubungan secara langsung sebagai berikut.

$$\delta_{pi} = f'_j(\text{net}_{pi}) \sum_k \delta_{pk} w_{kj}$$

Fungsi f harus differensiabel dan dengan fungsi sigmoid dinyatakan dalam fungsi sebagai berikut.

$$o_j = f\left(\sum_i w_{ji} v_i\right)$$

$$\frac{do_j}{d(\sum_i w_{ji} v_i)} = o_j (1 - o_j)$$

Nilai kesalahan akan menentukan perubahan bobot dalam persamaan berikut :

$$\Delta w_{ji} = \eta \delta_j o_i$$

Laju pembelajaran (*Learning rate* η) adalah parameter untuk menentukan seberapa cepat JST mencapai nilai target yang diinginkan (*Convergen*). Laju pembelajaran berupa konstanta

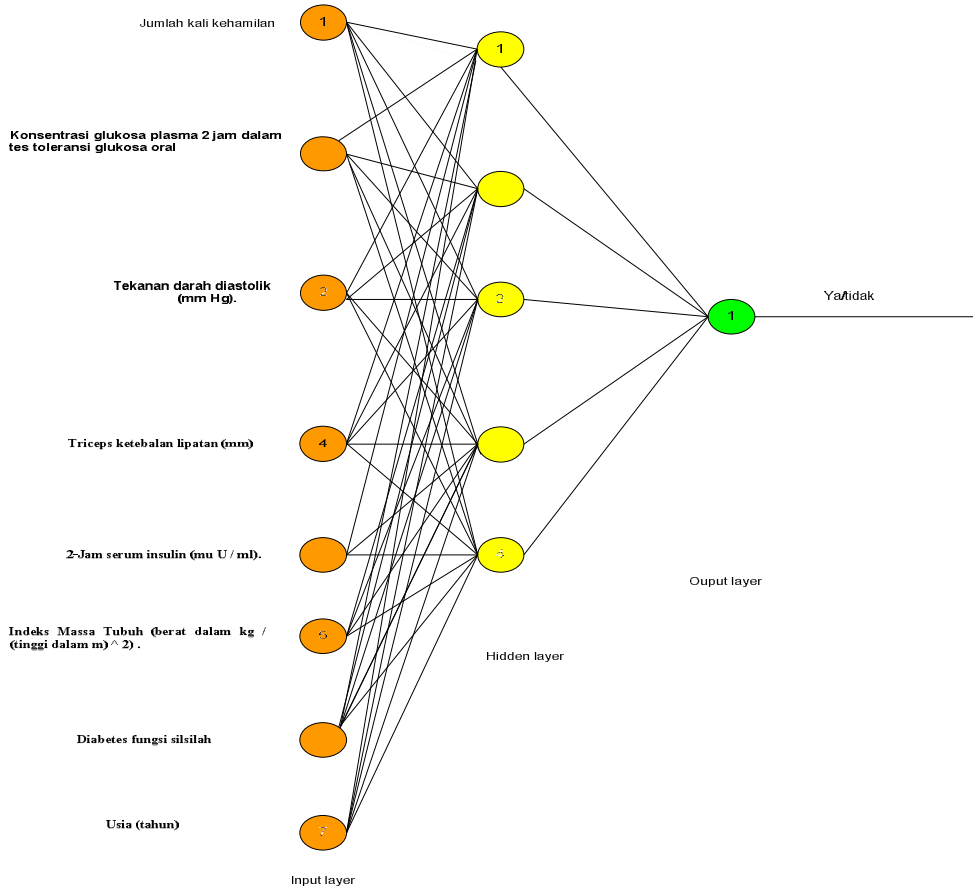
atau variabel yang berubah setiap kali terjadi perulangan. Bila laju ini terlalu besar, maka JST bisa terjebak ke dalam kondisi *Local Minima* yaitu suatu keadaan dimana JST mengira sudah konvergen dengan tingkat kesalahan pelatihan yang kecil, namun masih menghasilkan tingkat kesalahan pengujian yang besar. Bobot W dalam JST disimpan dan akan dipanggil kembali saat dibutuhkan untuk melakukan pengenalan pola.

2. METODE

Arsitektur Backpropagation Neural Network yang digunakan adalah sebagai berikut : (1) **8 input layer**, karena ada 8 kategori yang dijadikan sebagai data input untuk JST, yaitu : (a) Jumlah berapa kali hamil; (b) Konsentrasi glukosa plasma 2 jam dalam tes glukosa oral; (c) Tekanan darah diastolik (mm Hg); (d) Triceps ketebalan lipatan (mm); (e) 2-Jam serum insulin ($\mu\text{U} / \text{ml}$); (f) Indeks Massa Tubuh (berat dalam kg / (tinggi dalam m)²); (g) Diabetes fungsi silsilah; (h) Usia (tahun). (2) **2 output**

layer, yaitu output = 1, berarti orang tersebut menderita penyakit Diabetes, dan jika output = 0 maka orang tersebut tidak menderita Diabetes. (3) **5 Hidden layer**.

Dibawah ini adalah arsitektur *Backpropagation Neural Network*, untuk mendeteksi penyakit *Diabetes Mellitus* :



Gambar 3. Arsitektur *Backpropagation Neural Network*

3. HASIL DAN PEMBAHASAN

Berdasarkan data diabetes yang didapat dari *uci machine learning*

repository terdapat 768 data, dengan rincian 500 terkena diabetes dan 268 tidak terkena diabetes.

Langkah - Langkah Pengujian

: (1) Data dikonversi ke dalam format Excell. (2) Kemudian pisahkan data yang terkena diabetes dan tidak terkena kedalam 2 buah file data yang berbeda yaitu diabet.math dan nondiabet.math. (3) Membuat program pada matlab, Deklarasi nilai minimum dari semua data baik yang terkena diabetes maupun tidak terkena diabetes adalah 0 dan data maksimum adalah 846, Deklarasi range input adalah [0 846;0 846;0 846;0 846;0 846;0 846;0 846;0 846;0 846;0 846]; Training dilakukan sebanyak 8

kali. (4) Proses pelatihan : (a) Bentuk jaringan dengan perintah newff; (b) Minimalkan kuadrat kesalahan rata-rata (mse). Untuk meminimalkan kuadrat kesalahan rata-rata (MSE=mean square error) dilakukan proses pelatihan dengan menggunakan fungsi TRAIN; (c) Ubah bobot dengan metode penurunan gradient (*gradient descent*). Bobot dan bias diubah pada arah dimana unjuk kerja fungsi menurun paling cepat, yaitu dalam arah negatif gradientnya.

Pengujian data penderita diabetes :

```
>NN_diabet
```

```
Warning: NEWFF used in an obsolete way.
```

```
> In nntobsu at 18
```

```
  In newff at 105
```

```
  In NN_dia at 4
```

```
    See help for NEWFF to update calls to the new argument list.
```

```
TRAINLM-calcjx, Epoch 0/100, MSE 5.10953/0, Gradient 115.311/1e-010
```

```
TRAINLM-calcjx, Epoch 12/100, MSE 3.12001e-032/0, Gradient 1.88931e-014/1e-010
```

```
TRAINLM, Minimum gradient reached, performance goal was not met.
```

```
TRAINLM-calcjx, Epoch 0/100, MSE 0.208056/0, Gradient 11.41/1e-010
```

```
TRAINLM-calcjx, Epoch 4/100, MSE 2.46519e-032/0, Gradient 2.09935e-014/1e-010
```

```
TRAINLM, Minimum gradient reached, performance goal was not met.
```

```
TRAINLM-calcjx, Epoch 0/100, MSE 0.233202/0, Gradient 6.12821/1e-010
```

TRAINLM-calcjx, Epoch 8/100, MSE 1.57772e-028/0, Gradient 4.35105e-014/1e-010

TRAINLM, Minimum gradient reached, performance goal was not met.

TRAINLM-calcjx, Epoch 0/100, MSE 0.525498/0, Gradient 2.12229/1e-010

TRAINLM-calcjx, Epoch 3/100, MSE 6.55034e-024/0, Gradient 5.13463e-012/1e-010

TRAINLM, Minimum gradient reached, performance goal was not met.

TRAINLM-calcjx, Epoch 0/100, MSE 0.999994/0, Gradient 2/1e-010

TRAINLM-calcjx, Epoch 3/100, MSE 1.55585e-026/0, Gradient 2.49468e-013/1e-010

TRAINLM, Minimum gradient reached, performance goal was not met.

TRAINLM-calcjx, Epoch 0/100, MSE 0.499987/0, Gradient 2.44943/1e-010

TRAINLM-calcjx, Epoch 16/100, MSE 2.50371e-030/0, Gradient 3.65473e-015/1e-010

TRAINLM, Minimum gradient reached, performance goal was not met.

TRAINLM-calcjx, Epoch 0/100, MSE 0.501604/0, Gradient 2.45047/1e-010

TRAINLM-calcjx, Epoch 8/100, MSE 1.19248e-022/0, Gradient 2.18412e-011/1e-010

TRAINLM, Minimum gradient reached, performance goal was not met.

TRAINLM-calcjx, Epoch 0/100, MSE 0.927479/0, Gradient 37.1662/1e-010

TRAINLM-calcjx, Epoch 25/100, MSE 0.25/0, Gradient 2.32763e-008/1e-010

TRAINLM-calcjx, Epoch 50/100, MSE 0.248806/0, Gradient 0.445623/1e-010

TRAINLM-calcjx, Epoch 58/100, MSE 2.08642e-025/0, Gradient 9.13548e-013/1e-010

TRAINLM, Minimum gradient reached, performance goal was not met.

Untuk menampilkan bobot

```
net.IW{1,1}-->
```

```
ans =
```

Columns 1 through 5

```
0.0158 0.1654 0.0003 0.0112 0.0960
0.0007 -0.0340 -0.1047 -0.0462 0.0214
```

0.1165	-0.3940	-1.1195	-0.5884	-0.6106
0.0020	0.0060	-0.1131	0.0322	0.0623
0.0066	0.1148	0.0334	0.0202	-0.0012

Columns 6 through 8

0.0047	0.0015	0.0510
-0.0408	-0.0024	0.0044
-0.5615	-0.0003	-0.2233
0.1685	0.0059	-0.0036
0.0210	0.0019	0.0380

net.LW{2,1}-->

ans =

0.7282	0.5000	0.3344	-0.3590	0.6530
--------	--------	--------	---------	--------

Untuk menampilkan bias

net.b{1}-->

ans =

-0.7019
5.1598
1.7061
1.8065
-3.5481

net.b{2}-->

ans =

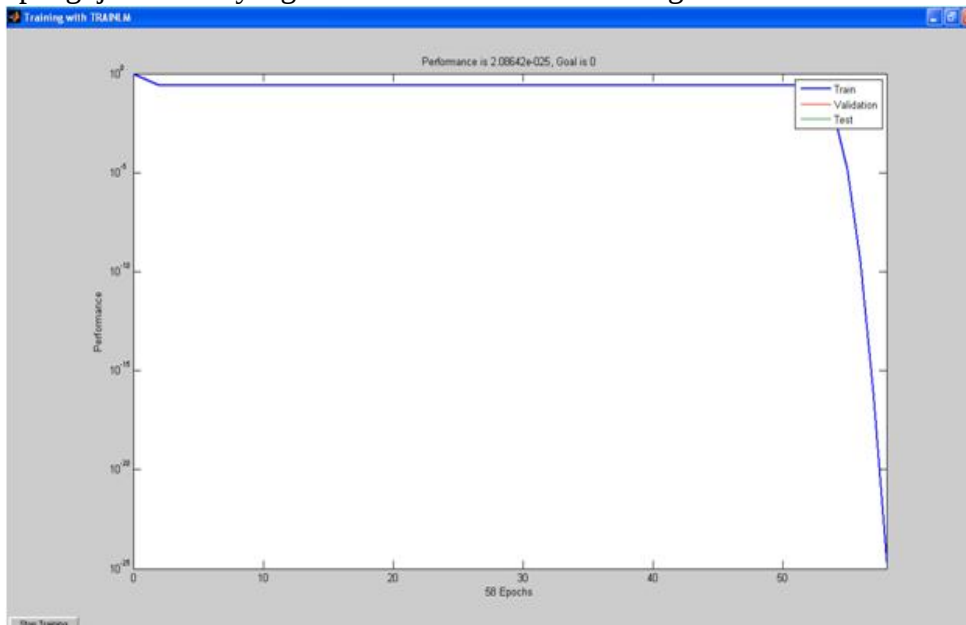
-0.1877

Untuk menampilkan output

a =

1

Hasil pengujian untuk yang terkena diabetes adalah sebagai berikut :



Gambar 4. Grafik pengujian terkena Diabetes Militus

Grafik pada Gambar 4 adalah grafik yang menunjukkan perubahan error . Menunjukkan bahwa iterasi diselesaikan dalam 50 epoh, berarti unjuk kerja (performance) yang diinginkan telah tercapai, yaitu pada saat nilai mse mendekati 0

Pengujian data bukan penderita diabetes :

```
>> NN_nondiabet
```

```
Warning: NEWFF used in an obsolete way.
```

```
> In nntobsu at 18
```

```
  In newff at 105
```

```
  In NN_dia at 4
```

```
  See help for NEWFF to update calls to the new argument list.
```

```
TRAINLM-calcjx, Epoch 0/100, MSE 0.466154/0, Gradient 81.5051/1e-010
```

```
TRAINLM-calcjx, Epoch 7/100, MSE 8.01187e-032/0, Gradient 5.98964e-015/1e-010
```

```
TRAINLM, Minimum gradient reached, performance goal was not met.
```

TRAINLM-calcjx, Epoch 0/100, MSE 0.0210089/0, Gradient 6.01145/1e-010
 TRAINLM-calcjx, Epoch 6/100, MSE 0/0, Gradient 0/1e-010
 TRAINLM, Performance goal met.

TRAINLM-calcjx, Epoch 0/100, MSE 0.146197/0, Gradient 34.3365/1e-010
 TRAINLM-calcjx, Epoch 6/100, MSE 6.16298e-033/0, Gradient 8.46617e-015/1e-010
 TRAINLM, Minimum gradient reached, performance goal was not met.

TRAINLM-calcjx, Epoch 0/100, MSE 1.05645/0, Gradient 58.6675/1e-010
 TRAINLM-calcjx, Epoch 7/100, MSE 1.68188e-029/0, Gradient 8.50444e-014/1e-010
 TRAINLM, Minimum gradient reached, performance goal was not met.

TRAINLM-calcjx, Epoch 0/100, MSE 0.00845768/0, Gradient 8.47059/1e-010
 TRAINLM-calcjx, Epoch 3/100, MSE 1.24862e-029/0, Gradient 1.48496e-014/1e-010
 TRAINLM, Minimum gradient reached, performance goal was not met.

TRAINLM-calcjx, Epoch 0/100, MSE 0.436514/0, Gradient 85.7332/1e-010
 TRAINLM-calcjx, Epoch 8/100, MSE 2.46519e-032/0, Gradient 3.50084e-015/1e-010
 TRAINLM, Minimum gradient reached, performance goal was not met.

TRAINLM-calcjx, Epoch 0/100, MSE 0.512484/0, Gradient 2.49566/1e-010
 TRAINLM-calcjx, Epoch 14/100, MSE 5.57133e-030/0, Gradient 1.84684e-014/1e-010
 TRAINLM, Minimum gradient reached, performance goal was not met.
 TRAINLM-calcjx, Epoch 0/100, MSE 0.956927/0, Gradient 4.11892/1e-010
 TRAINLM-calcjx, Epoch 22/100, MSE 1.47739e-027/0, Gradient 1.35759e-011/1e-010
 TRAINLM, Minimum gradient reached, performance goal was not met.

Untuk menampilkan bobot

net.IW{1,1}-->

ans =

Columns 1 through 5

-0.0085	-0.3756	-0.0389	-0.1270	-0.2613
-0.0465	-2.3188	-1.6617	-0.7898	-4.4186
-0.0010	-0.0269	-0.0239	-0.0008	-0.0226
-0.0005	-0.0385	-0.0146	-0.0005	-0.0147

-0.0014 0.0144 0.0189 0.0231 -0.0120

Columns 6 through 8

-0.2279 -0.0055 -0.1138

-0.7489 -0.0145 -0.5004

0.0097 0.0006 -0.0131

-0.0120 -0.0001 -0.0078

0.0230 -0.0004 -0.0086

net.LW{2,1}-->

ans =

0.1384 0.4355 -0.6055 -0.2493 -1.2298

Untuk menampilkan bias

net.b{1}-->

ans =

1.3254

-0.6980

0.7631

-0.1907

-2.4476

net.b{2}-->

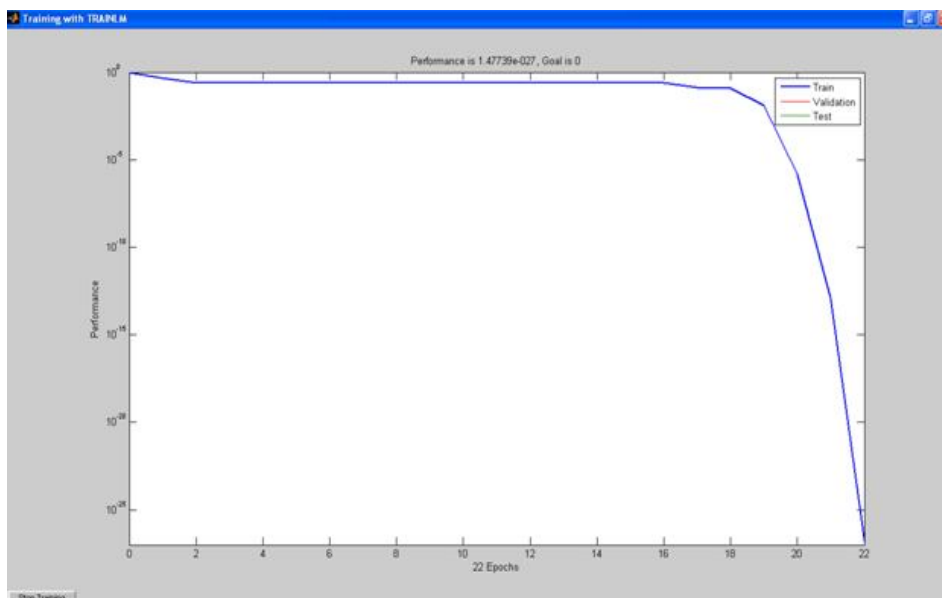
ans =

-0.5107

Untuk menampilkan output

a =

0



Gambar 5. Grafik pengujian tidak terkena Diabetes

Pada Gambar 5 adalah grafik yang menunjukkan pola perubahan error, dimana iterasi diselesaikan dalam 22 epoh, berarti unjuk kerja (performance) yang diinginkan telah tercapai, yaitu pada saat nilai mse mendekati 0

4. KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan hasil percobaan maka dapat diambil kesimpulan sebagai berikut : (1) Arsitektur Backpropagation Neural Networks yang digunakan memiliki 8 input layer

sesuai dengan jumlah properti data, 2 output layer berdasarkan klasifikasi data (terkena diabetes dan tidak terkena diabetes) dan 5 hidden layer. (2) Metode Backpropagation Neural Networks berhasil mengklasifikasikan data yang terkena diabetes dan yang tidak terkena diabetes. (3) Tingkat akurasi yang dihasilkan mendekati 100% pada training sebanyak 8 kali.

Saran

Saran yang dapat diberikan adalah : (1) Memperbanyak jumlah input layer agar diperoleh hasil yang

lebih akurat. (2) Memperbanyak jumlah klasifikasi data untuk bisa mendeteksi jenis-jenis penyakit diabetes

DAFTAR PUSTAKA

- Fausett, Laurene V., *“Fundamentals of Neural Network: Architectures, Algorithms, and Applications”*, Prentice-Hall, 1993.
- Marimin, *“Teori dan Aplikasi Sistem Pakar dalam teknologi manajerial”*, IPB press, Bogor, 2009
- Siang, Jong Jek, *“Jaringan syaraf tiruan dan pemrogramannya menggunakan Matlab”*, ANDI Offset, Jogjakarta, 2009
- _____, *Definition, Diagnosis and Classification of Diabetes Mellitus and Its Complications*, WHO, 1999
- <http://archive.ics.uci.edu/ml/dataset/pima+indian+diabetes>