PENERAPAN ALGORITMA *MULTILAYER PERCEPTRON* UNTUK DETEKSI DINI PENYAKIT DIABETES

Ahmad Setiadi

Jurusan Manajemen Informatika
Akademi Manajemen Informatika dan Komputer Bina Sarana Informatika Karawang
Jl. Banten No. 1 Karangpawitan, Karawang Telp : (0267)8454893
E-Mail : ahmad.ams@bsi.ac.id

Abstract

Each year, patient of diabetes mellitus is increasing, so that is needed the diagnose technique which is effective to detect in early. Neural network as a model of data mining can be used to predict wether someone is suffered from diabetes mellitus or not. In this research, Multilayer Perceptron (MLP) as a neural network algorithm is used, not only because this algorithm has a good ability in predicting but also because this algorithm is commonly used. In this research, the processed data is total of 768 records and as a result of checking up Indian Pima women at least 21 years old.. To implement the MLP algorithm, SPSS Neural Network 17.0 is used. The result of implementing algorithm then is evaluated by using confusion matrix method and ROC (Receiver Operating Characteristic) curve method. This result is proved that implementation of MLP algorithm to detect diabetes mellitus for Indian Pima has a good performance. The value of accuracy by confusion matrix method is 77,7 %. Using ROC curve method, this research shows the accuracy of 0,83, so that it is including as good classification because it is being among 0,8 until 0,9. This research proved that MLP Algorithm can be used to detect diabetes mellitus in early time.

Keywords: *diabetes mellitus, neural network model, multilayer perceptron, confusion matrix, kurva ROC*

1. PENDAHULUAN

Penyakit diabetes mellitus dikenal juga sebagai penyakit gula atau kencing manis. Jenis penyakit ini merupakan suatu penyakit yang disebabkan oleh adanya gangguan menahun terutama pada sistem metabolisme karbohidrat, lemak, dan juga protein dalam tubuh. Gangguan metabolisme tersebut disebabkan kurangnya hormon insulin, yang diperlukan dalam proses pengubahan gula menjadi tenaga serta sintesis lemak (Lanywati, 2001).

Kondisi demikian mengakibatkan terjadinya *hiperglikemia*, yaitu meningkatnya kadar gula dalam darah atau terdapatnya kandungan gula dalam air kencing dan zat-zat keton serta asam (*keto-acidosis*) yang berlebihan. Keberadaan zat-zat keton dan asam yang berlebihan ini menyebabkan terjadinya rasa haus yang terus menerus, banyak kencing, penurunan berat badan meskipun selera makan tetap baik, serta penurunan daya tahan tubuh yang menyebabkan tubuh menjadi lemah dan mudah sakit. Penderita diabetes tidak jarang mengalami kematian pada usia muda(Lanywati, 2001).

Penyakit diabetes mellitus merupakan salah satu penyakit yang paling banyak diidap. Menurut data WHO (*World Health Organization*) tahun 1990, lebih kurang 2% dari total penduduk dunia merupakan penderita diabetes mellitus (Lanywati, 2001). Di Indonesia, jumlah penyandang diabetes mellitus semakin menunjukkan peningkatan yang sangat tinggi. Pada tahun 2000, jumlah penyandang di Indonesia sebanyak 8,4 juta jiwa dan diperkirakan akan mencapai angka 21,3 juta jiwa pada tahun 2030 nanti. Hal tersebut mengakibatkan Indonesia berada di peringkat keempat jumlah penyandang diabetes mellitus di dunia setelah Amerika Serikat, India, dan Cina (Kusumadewi, 2009). Tahun 2008, diabetes mellitus menempati peringkat 10 besar penyebab kematian di Indonesia, yaitu menempati peringkat keenam dengan 5,7% (Departemen Kesehatan RI, 2009).

Jumlah penderita diabetes mellitus di Indonesia yang tergolong tinggi, dan perkiraan adanya peningkatan di tahun-tahun mendatang menyebabkan perlunya antisipasi dan tindakan segera dalam pendeteksian penyakit ini sedini mungkin . Hal tersebut dimaksudkan agar penanganan terhadap penderita dapat dilakukan dengan cepat. Oleh karenanya, dibutuhkan suatu teknik analisa yang efisien dan efektif untuk mendiagnosa penyakit ini.

Saat ini banyak data medis pasien yang dimiliki oleh institusi kesehatan. Dalam data medis tersebut terdapat banyak informasi tersembunyi yang dapat dimanfaatkan, seperti

penggunaan data medis untuk mendiagnosa penyakit yang diderita oleh pasien. Berdasarkan data medis tersebut, seorang pasien dapat didiagnosa apakah terdetiksi menderita suatu penyakit atau tidak. Dengan pemanfaatan data medis tersebut, keputusan medis, apakah seseorang terdeteksi menderita penyakit diabetes atau tidak, akan dapat diperoleh secara efektif dan efisien.

Diagnosa penyakit dapat dilakukan dengan menggunakan teknik konvensional, namun saat ini hal tersebut tidak lagi dirasakan efektif. Hal ini dikarenakan sudah dimungkinkannya penggunaan sistem berbasis komputer sebagai teknik analisa dalam mendiagnosa penyakit. Salah satu pemanfaatan sistem berbasis komputer untuk mendiagnosa penyakit adalah penggunaan *data mining*.

Dalam *data mining* dikenal banyak model yang dapat digunakan untuk mendiagnosa penyakit. salah satu model yang banyak digunakan untuk adalah model *neural network*. Model ini banyak digunakan karena memiliki kemampuan untuk memprediksi dengan baik. Model *neural network* ini juga memiliki banyak algoritma dengan tingkat kemampuan prediksi yang berbeda. Salah satu algoritma yang memiliki tingkat keakuratan yang tinggi adalah algoritma *Multilayer Perceptron* (MLP).

MLP merupakan algoritma yang sangat terkenal dan paling banyak digunakan pada penelitian-penelitian di bidang kesehatan, teknik, model matematika dan lain-lain. Bahkan, Sekitar 95% aplikasi bisnis yang menggunakan *neural network*, memakai algoritma ini (Vercellis, 2009).

Untuk menerapkan algoritma MLP ini digunakan perangkat lunak SPSS *Neural Network* 17.0. Perangkat lunak ini dipilih mengingat SPSS *Statistics* 17.0 merupakan perangkat lunak yang memiliki sistem yang komprehensif untuk analisa data. Selain itu, perangkat lunak SPSS *Neural network* ini banyak digunakan karena kemampuan, fleksibilitas dan kemudahan dalam penggunaannya.

Data yang digunakan dalam penelitian ini adalah data sekunder yang bersumber dari alamat web: http://archive.ics.uci.edu/ml/. Data yang diteliti merupakan hasil pemeriksaan terhadap 768 pasien berjenis kelamin perempuan berusia minimal 21 tahun yang memiliki keturunan suku Indian Pima, dimana 268 orang (34,9%) terdeteksi positif menderita diabetes mellitus dan 500 orang (65,1%) negatif menderita diabetes mellitus.

Hasil dari penerapan model *neural network* menggunakan algoritma MLP ini kemudian akan diukur tingkat akurasinya menggunakan metode *Confusion matrix* dan Kurva ROC (*Receiver Operating Characteristic*) untuk mengetahui apakah algoritma MLP memiliki tingkat akurasi yang baik, sehingga tujuan penerapan algoritma MLP untuk prediksi penyakit diabetes ini dapat tercapai.

2. STUDI PUSTAKA

a. Tinjauan Studi

Bendi Venkata Ramana dan M. Surendra Prasad Babu melakukan penelitian untuk meneliti penerapan beberapa model *data mining* untuk mendiagnosa penyakit hati (*liver diseas*). Data yang digunakan dalam penelitian ini berasal dari 583 orang (416 orang terdeteksi menderita penyakit hati dan 167 orang tidak terdeteksi). Penelitian ini menghasilkan tingkat akurasi untuk algoritma MLP sebesar 74,7826% (Ramana & Babu, 2012)

J. Padmavathi melakukan penelitian untuk mengukur tingkat akurasi penerapan beberapa algoritma *neural network* untuk prediksi penyakit kanker payudara. Data yang digunakan dalam penelitian ini berjumlah 580 *record*. Penelitian ini menghasilkan tingkat akurasi untuk algoritma MLP sebesar 91,3% (Padmavathi, 2011).

P. Venkatesan dan S. Anitha melakukan penelitian untuk mengukur tingkat akurasi penerapan beberapa algoritma model *neural network* untuk diagnosa penyakit diabetes mellitus. Data yang digunakan untuk penelitian berasal dari 1200 orang (600 orang penderita diabetes mellitus dan 600 non penderita diabetes mellitus) yang datang ke rumah sakit dalam kurun waktu tahun 1996 sampai 1998. Kriteria yang digunakan didasarkan pada kriteria yang ditetapkan oleh WHO, yaitu *Fasting Plasma Glukose* (FPG). Pada penelitian tersebut dihasilkan tingkat akurasi prediksi untuk algoritma MLP sebesar 91,3% (Venkatesan & Anitha, 2006).

b. Tinjauan Pustaka

1. ***Data Mining***

Data mining telah menarik banyak perhatian dalam dunia sistem informasi dan masyarakat secara keseluruhan dalam beberapa tahun terakhir. Hal ini dikarenakan tersedianya data dalam jumlah besar dan adanya kebutuhan untuk segera mengubah data tersebut menjadi informasi dan pengetahuan yang berguna. Informasi dan pengetahuan yang diperoleh dapat digunakan untuk banyak aplikasi, mulai dari analisa pasar, deteksi penipuan, dan retensi pelanggan, untuk pengendalian produksi dan

eksplorasi ilmu pengetahuan (Han & Kamber, 2007).

Data mining, sering juga disebut *Knowledge Discovery in Database* (KDD), adalah kegiatan yang meliputi pengumpulan dan pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar (Santosa, 2007).

Data mining melibatkan statistik, teknologi *database*, pengenalan pola (*pattern recognition*), pembelajaran berbasis mesin (*machine learning*), visualisasi data (*data visualization*) dan sistem pakar (*expert system*). Database adalah kumpulan data yang diorganisasikan sedemikian rupa, sehingga data yang ada di dalamnya dapat dengan mudah diakses, diatur dan diperbaharui (*update*). *Database* berisi kumpulan record-record dan file-file, dan *database manager* menyediakan kemampuan pengendalikan akses untuk merekam dan membaca, menspesifikasikan pembuatan laporan (*report generation*) dan penggunaan data untuk analisa. (Obenshain, 2004)

Berdasarkan tugas yang dijalankannya, *data mining* dapat dikelompokkan menjadi enam (Larose, 2005), yaitu:

a) Deskripsi
   Deskripsi adalah mencari cara untuk menggambarkan pola (*pattern*) dan *trend* yang terdapat dalam data.
b) Estimasi
   Estimasi mirip dengan klasifikasi, namun variabel sasarannya adalah numerik. Model dibuat menggunakan *record* yang lengkap dan juga ada variabel targetnya. Kemudian untuk data baru, estimasi nilai variabel target dibuat berdasarkan nilai *predictor*.
c) Prediksi
   Prediksi mirip seperti klasifikasi dan estimasi, tetapi hasilnya untuk memprediksi di masa depan. Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan untuk prediksi (dalam keadaan yang tepat).
d) Klasifikasi
   Pada klasifikasi, yang menjadi sasaran adalah variabel kategori, misalnya atribut penghasilan, yang bisa dikategorikan menjadi tiga kelas atau kategori, yaitu tinggi, sedang, dan rendah. Model *data mining* membaca sejumlah besar *record,* dimana tiap *record* berisi informasi pada variabel target.
e) Clustering
   Clustering mengacu pada pengelompokkan *record-record*, observasi, atau kasus-kasus ke dalam kelas-kelas dari objek yang mirip. Pada *clustering* tidak ada variabel sasaran. Sebuah *cluster* adalah koleksi *record* yang mirip satu sama lain, dan tidak mirip dengan *record* pada *cluster*. Tidak seperti klasifikasi, pada *clustering* tidak ada variabel target.
f) Asosiasi
   Tugas asosiasi untuk *data mining* adalah kegiatan untuk mencari atribut yang "*go together*". Dalam dunia bisnis, asosiasi dikenal sebagai *affinity analysis* atau *market basket analysis*, tugas asosiasi adalah membuka *rules* untuk pengukuran hubungan antara dua atribut atau lebih.

2. ***Neural Network***

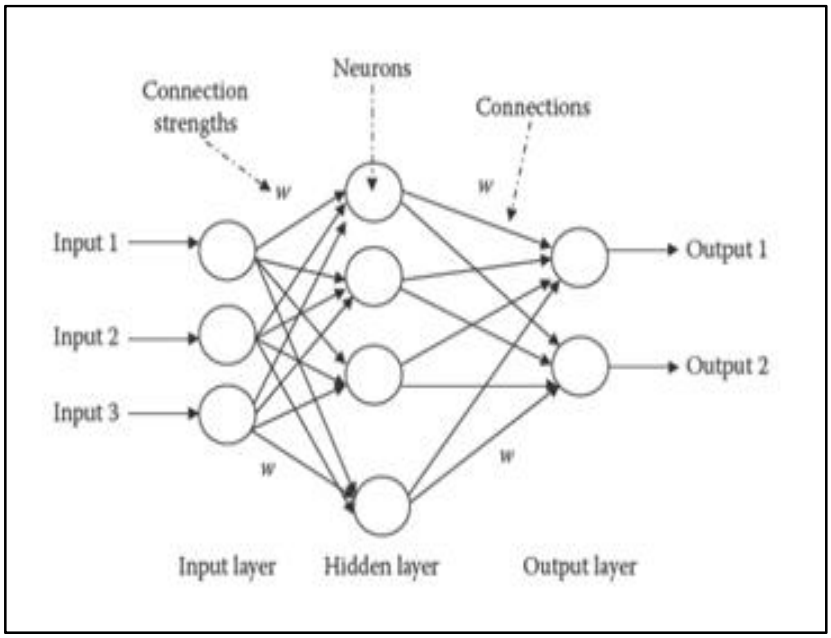
Neural network adalah sistem pengolah informasi yang mempunyai karakteristik mirip dengan jaringan syaraf biologis, yaitu jaringan syaraf pada otak manusia. Karakteristik dari *neural network* ditentukan oleh beberapa hal, yaitu (Noertjahyana & Yulia, 2002):

a. Arsitektur Jaringan
   Arsitektur, merupakan bentuk pola hubungan antara neuron-neuronnya. Pada *neural network*, neuron-neuron tersusun dalam *layer* (lapisan*)*. Pengaturan neuron dalam *layer* dan hubungan-hubungannya disebut dengan arsitektur jaringan. *Neural network* dapat diklasifikasikan menjadi dua jenis, yaitu *single layer* dan *multilayer*. Dalam jaringan *single layer*, neuron-neuron dapat dikelompokkan menjadi dua bagian, yaitu *input units* (unit-unit masukan) dan *output units* (unit-unit keluaran). Sedangkan dalam jaringan *multilayer*, selain ada unit-unit *input* dan *output* juga terdapat *hidden units* (unit-unit tersembunyi).
b. Pembelajaran (*Learning*) *Neural Network*
   Learning algorithm (algoritma proses pembelajaran), merupakan metode yang digunakan untuk menentukan bobot dari hubungannya. Tujuan dari pelatihan *neural network* adalah untuk mencari bobot-bobot yang terdapat dalam tiap *layer*. Ada dua jenis pelatihan dalam sistem *neural network*, yaitu proses belajar terawasi (*supervised learning*) dan proses belajar tak terawasi (*unsupervised learning*).
   Dalam proses belajar yang terawasi, seolah-olah ada "guru" yang mengajari *neural network*. Cara pelatihan *neural network* ini adalah dengan memberikan data-data yang disebut *training data* atau *training vectors*, yang terdiri dari pasangan input dan output yang diinginkan dan kemudian diberikan ke *neural network* sehingga *neural network* dapat memodifikasi bobot-bobot yang ada untuk mencoba mencari kesamaan antara

hasil *output* yang dihasilkan oleh *neural network* dengan hasil *output* yang diinginkan. Setelah proses pelatihan selesai, *neural network* kemudian diberi suatu nilai *input* dan akan menghasilkan suatu hasil *output*. Dalam proses belajar yang tak terawasi, tidak ada "guru" yang mengajari. *Neural network* hanya diberi data *input* (*input vectors*) dan tidak dilengkapi dengan suatu hasil *output* yang diinginkan. *Neural network* akan memodifikasi bobot sehingga untuk *input* yang hampir sama, *output* yang dihasilkan sama.

c. Fungsi Aktivasi
   Fungsi aktivasi, merupakan fungsi untuk menghasilkan *output*. Fungsi aktivasi yang biasa digunakan, antara lain:
   1) Fungsi Identitas
   2) Fungsi *Sigmoid Biner*
   3) Fungsi *Sigmoid Bipolar*
   4) Fungsi *Hyberbolic Tangent*
   5) Fungsi *Softmax*

d. Pelatihan *Neural Network*
   Selain untuk meminimalkan *error* pada *output* yang dihasilkan oleh jaringan, tujuan lain dari pelatihan *neural network* adalah memperoleh keseimbangan antara kemampuan untuk merespon *input pattern* yang digunakan pada proses pelatihan secara benar, hal ini dikenal dengan *memorization* dan kemam puan untuk memberikan respon yang masuk akal (baik) terhadap *input* yang mirip, tetapi tidak sama persis dengan yang digunakan pada saat *training* (*generalization*).



Gambar 1. Arsitektur *Neural Network*
Sumber: Kusrini & Luthfi (2009)

3. **Algoritma *Mutilayer Perceptron* (MLP)**

Algoritma MLP merupakan algoritma yang mengadopsi cara kerja jaringan saraf pada makhluk hidup. Algoritma ini terkenal handal karena proses pembelajaran yang mampu dilakukan secara terarah. Pembelajaran algoritma ini dilakukan dengan peng*update*an bobot balik (*back propagation*). Penetapan bobot yang optimal akan berujung pada hasil prediksi yang tepat.

Pada MLP, digunakan fungsi standar *Sigmoid* dimana jumlah pembobotan dari sejumlah *input* dan bias dimasukkan ke *activation level* melalui fungsi transfer untuk menghasilkan *output*, dan unit-unit diatur dalam lapisan topologi *feed-forward* yang disebut *Feed Forward Neural Network* (FFNN) (Venkatesan & Anitha, 2006).

Ketika ada lebih dari satu lapisan tersembunyi (*hidden layer*), keluaran (*output*) dari lapisan tersembunyi dimasukkan ke *hidden layer* berikutnya dan bobot terpisah digunakan untuk penjumlahan ke setiap lapisan berikutnya.

MLP terdiri dari *input layer*, satu atau lebih *hidden layer*, dan *output layer*. Berikut penjelasan masing-masing layer: (Vercellis, 2009)

a. *Input layer*
   Input layer untuk menerima nilai masukan dari tiap *record* pada data. Jumlah simpul *input* sama dengan jumlah variabel prediktor.
b. *Hidden layer*
   Hidden layer mentransformasikan nilai *input* di dalam *network*. Tiap simpul pada *hidden layer* terhubung dengan simpul-simpul pada *hiden layer* sebelumnya atau dari simpul-simpul pada *input layer* dan ke simpul-simpul pada *hidden layer* berikutnya atau ke simpul-simpul pada *output layer*. Jumlah *hidden layer* bisa berapa saja.
c. *Output layer*
   Garis yang terhubung dengan *Output layer* berasal dari *hidden layer* atau *input layer* dan mengembalikan nilai keluaran yang bersesuaian dengan variabel prediksi. Keluaran dari *output layer* biasanya merupakan nilai *floating* antara 0 sampai 1 (Kusrini & Luthfi, 2009).

Setiap simpul dalam *neural network* merupakan sebuah unit pemrosesan. Tiap simpul memiliki beberapa masukan dan sebuah keluaran. Setiap simpul mengkombinasikan beberapa nilai masukan, melakukan kalkulasi, dan membangkitkan nilai keluaran (aktivasi). Dalam setiap simpul terdapat dua fungsi, yaitu fungsi untuk mengkombinasikan masukan dan fungsi aktifasi untuk menghitung keluaran. Terdapat beberapa metode untuk mengkombinasikan masukan antara lain *weighted sum*, *mean*, *max*, logika *OR*, atau logika *AND*. Untuk fungsi aktivasi juga terdapat beberapa metode seperti, fungsi *heaviside* (*threshold*), *piecewise*, *gaussian*, *sigmoid* (*logistic*), *hyberbolic tangent*, *sine and cosine*, *linear* (*identity*) (Gorunescu, 2011).

Backpropagation bekerja melalui proses secara iteratif menggunakan data *training*, membandingkan nilai prediksi dari jaringan dengan setiap data yang terdapat pada data *training*. Dalam setiap proses, bobot relasi dalam jaringan dimodifikasi untuk meminimalkan nilai *Mean Squared Error* (MSE) antara nilai prediksi dari *network* dengan nilai yang sesungguhnya. Modifikasi relasi *neural network* tersebut dilakukan dengan arah mundur, dari *output layer* hingga *layer* pertama dari *hidden layer* sehingga algoritma ini disebut *backpropagation.*

Langkah pembelajaran dalam algoritma *backpropagation* adalah sebagai berikut (Myatt, 2007):

1) Inisialisasi bobot jaringan secara acak (biasanya antara -0.1 sampai 1.0)
2) Untuk setiap data pada data *training*, hitung *input* untuk simpul berdasarkan nilai *input* dan bobot jaringan saat itu, menggunakan rumus:

$$Input_j = \sum_{i=1}^{n} O_i w_{ij} + \bar{\theta}_j$$

Keterangan:
O_i = *Output* simpul *i* dari layer sebelumnya
w_{ij} = bobot relasi dari simpul *i* pada *layer* sebelumnya ke simpul *j*
θ_j= bias (sebagai pembatas)

3) Berdasarkan *input* dari langkah dua, selanjutnya membangkitkan *output* untuk simpul menggunakan fungsi aktifasi *sigmoid*:

$$Output = \frac{1}{1 + e^{-Input}}$$

4) Hitung nilai *Error* antara nilai yang diprediksi dengan nilai yang sesungguhnya menggunakan rumus:

$$Error_j = Output_j \cdot (1 - Output_j) \cdot (Target_j - Output_j)$$

Keterangan:
$Output_j$ = *Output* aktual dari simpul *j*
$Target_j$ = Nilai target yang sudah diketahui pada data *training*

5) Setelah nilai *Error* dihitung, selanjutnya dibalik ke *layer* sebelumnya (*backpropagated*). Untuk menghitung nilai *Error* pada *hidden layer*, menggunakan rumus:

$$Error_j = Output_j(1 - Output_j) \sum_{k=1}^{n} Error_k \, w_{jk}$$

Keterangan:
$Output_j$ = *Output* aktual dari simpul j
$Error_k$ = *error* simpul *k*
w_{jk} = Bobot relasi dari simpul *j* ke simpul *k* pada *layer* berikutnya

6) Nilai *Error* yang dihasilkan dari langkah sebelumnya digunakan untuk memperbarui bobot relasi menggunakan rumus

$$w_{ij} = w_{ij} + l \cdot Error_j \cdot Output_i$$

Keterangan:
w_{ij} = bobot relasi dari unit *i* pada *layer* sebelumnya ke unit *j*
l = *learning rate* (konstanta, nilainya antara 0 sampai dengan 1)
$Error_j$ = *Error* pada output layer simpul *j*
$Output_i$ = *Output* dari simpul *i*

4. **Metode Evaluasi dan Validasi pada *Data Mining***

Untuk menguji model pada *data mining* dapat digunakan beberapa metode, diantaranya adalah metode *Confusion Matrix* dan kurva ROC (*Receiver Operating Characteristic*).

a. Metode *Confusion Matrix*

Pada metode *confusion matrix*, jika data set hanya terdiri dari dua kelas, kelas yang satu dianggap sebagai positif dan yang lainnya negatif (Bramer, 2007). Sehingga pengklasifikasin data disusun dalam tabel 1 seperti berikut:

Tabel 1. Model *Confusion Matrix*

Klasifikasi yang benar	Diklasifikasikan Sebagai	
	+	-
+	*True positives*	*False negatives*
-	*False positives*	*True negatives*

Sumber : Bramer (2007)

Pada Tabel 1(model *confusion matrix*), data akan diklasifikasikan ke dalam empat jenis, yaitu dengan cara membandinagkan antara data kondisi yang sebenarnya dengan data yang didapat dari hasil prediksi melalui pengolahan data yang dibuat oleh model *neural network* (algoritma) yang digunakan. Empat klasifikasi tersebut adalah:

- *True Positif* : Data yang kondisi sebenarnya benar (positif) dan data hasil prediksi juga diklasifikasikan benar (positif).
- *False Positif* : Data yang kondisi sebenarnya salah (negatif), tetapi data hasil prediksi diklasifikasikan benar (positif).
- *False Negatif* : Data yang kondisi sebenarnyai salah (negatif) dan data hasil prediksi juga diklasidfikasikan salah (negatif).
- *True Negatif* : Data yang kondisi sebenarnya benar (positif), tetapi data hasil prediksi diklasifikasikan salah (negatif).

Setelah data diklasifikasikan, kemudian dapat diukur tingkat akurasinya dengan rumus:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Keterangan :
TP : *True Positif*
TN : *True Negatif*
FP : *False Positif*
FN : *False Negatif*

b. Metode Kurva ROC (*Receiver Operating Characteristic*)

Kurva ROC menunjukkan akurasi dan membandingkan klasifikasi secara visual. ROC mengekspresikan *confusion matrix*. ROC adalah grafik dua dimensi dengan *false positives* sebagai garis horisontal dan *true positives* sebagai garis vertikal (Vercellis, 2009). *Area Under Curve* (AUC) dihitung untuk mengukur perbedaan performansi metode yang digunakan. AUC dihitung menggunakan rumus: (Liao, 2007)

$$\theta^r = \frac{1}{mn}\sum_{j=1}^{n}\sum_{i=1}^{m}\psi\,(xi^r, xj^r)$$

Dimana $\psi(X,Y) = \begin{cases} 1 & Y < X \\ \frac{1}{2} & Y = X \\ 0 & Y > X \end{cases}$

K = jumlah algoritma klasifikasi yang dikomparasi
X = *output positif*
Y = *output negatif*

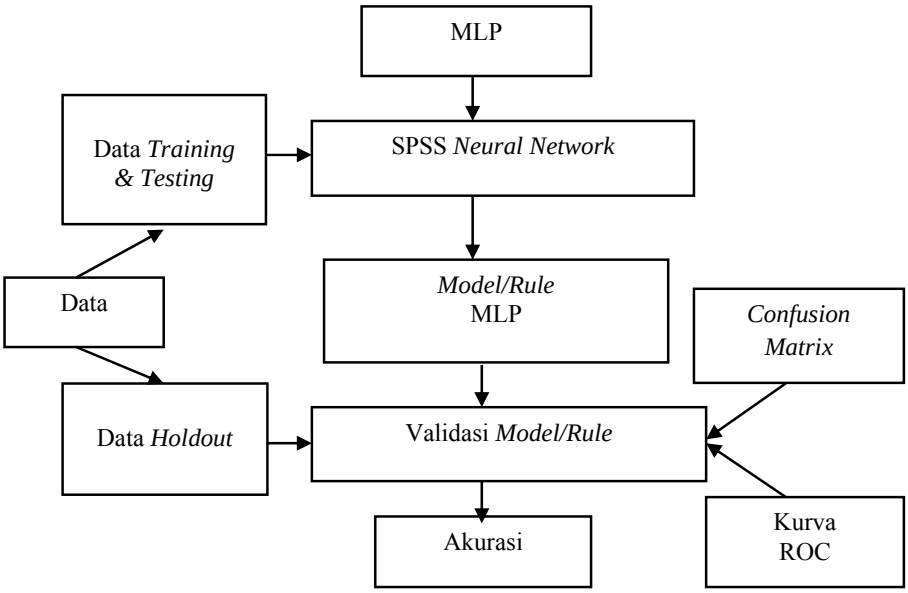
Pada *data mining*, nilai AUC dapat dibagi menjadi beberapa kelompok (Gorunescu, 2011), yaitu:

a. 0.90 sampai 1.00 : klasifikasi sangat baik (*excelent classification*)
b. 0.80 sampai 0.90 : klasifikasi baik (*good classification*)
c. 0.70 sampai 0.80 : klasifikasi cukup (*fair clasifcation*)
d. 0.60 sampai 0.70 : klasifikasi buruk (*poor classification*)
e. 0.50 sampai 0.60 : klasifikasi salah (*failure classification*)

3. METODE PENELITIAN

Jenis penelitian yang dilakukan dalam penelitian ini adalah jenis penelitian eksperimen. Metode ini menguji kebenaran sebuah hipotesis dengan statistik dan menghubungkannya dengan masalah penelitian (Kothari, 2004).

Penelitian ini dilaksanakan dengan melakukan beberapa langkah penelitian yang dilandasi oleh kerangka pemiiran pemecahan masalah sebagai berikut:



Gambar 2: Kerangka Pemikiran Pemecahan Masalah

4. PEMBAHASAN

a. Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data sekunder. Data bersumber dari alamat web: http://archive.ics.uci.edu/ml/. Data yang diperoleh merupakan hasil pemeriksaan terhadap 768 pasien berjenis kelamin perempuan berusia minimal 21 tahun yang memiliki keturunan suku Indian Pima. Parameter input (variabel *predictor*) dalam *neural network* ini terdiri dari delapan variabel *input* dan satu *output* yang dihasilkan adalah variabel Class (0 untuk negatif penderita diabetes atau 1 untuk positif penderita diabetes). Variabel atau atribut, tipe, ukuran dan nilai atribut telihat pada tabel berikut:

Tabel 2. Atribut, tipe, ukuran dan nilai atribut

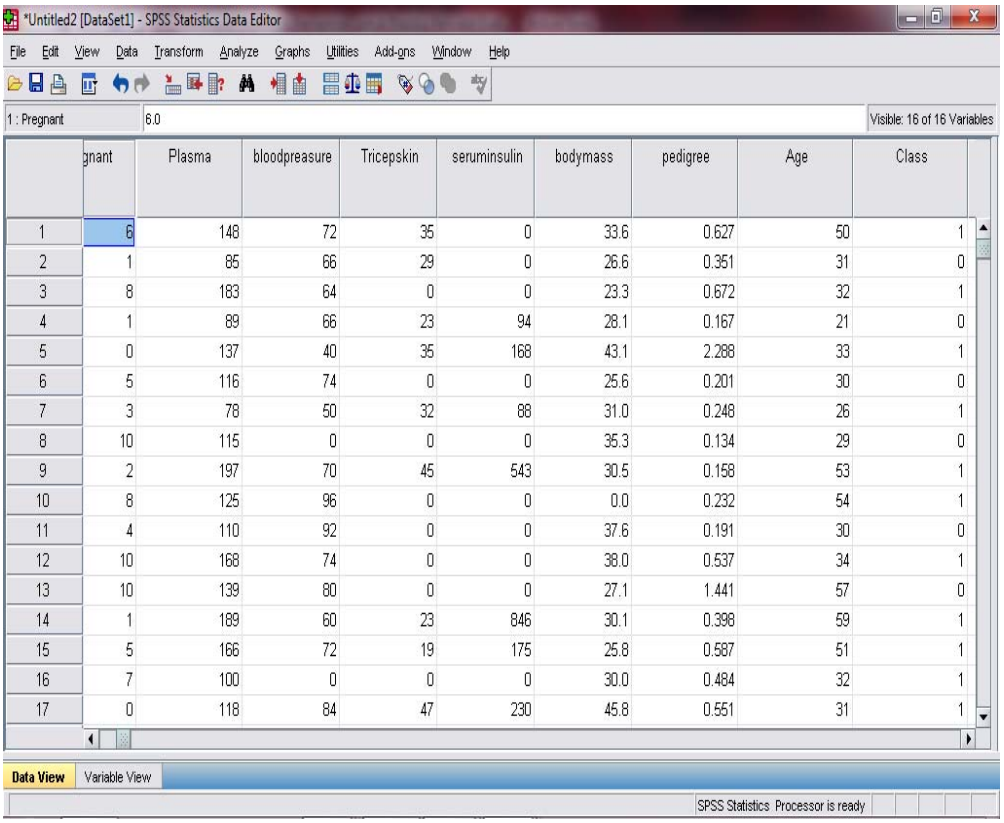
No	Atribut	Tipe	Ukuran	Nilai Atribut
1.	Pregnant	*Numeric*	*Scale*	Angka
2.	Plasma	*Numeric*	*Scale*	Angka
3.	Bloodpreasure	*Numeric*	*Scale*	Angka
4.	Tricepskin	*Numeric*	*Scale*	Angka
5.	Serum Insulin	*Numeric*	*Scale*	Angka
6.	Body Mass	*Numeric*	*Scale*	Angka

7.	Pedigree	*Numeric*	*Scale*	Angka
8.	Age	*Numeric*	*Scale*	Angka
9.	Class	*Numeric*	*Nominal*	0 (Negatif Diabetes) 1 (Positif Diabetes)

Sumber: Penulis

Pada tabel 2 terlihat bahwa terdapat sembilan atribut yang digunakan. Delapan atribut akan digunakan sebagai atribut *predictor* atau masukan (input), yaitu Pregnant, Plasma, Bloodpreasure, Tricepskin, Serum Insulin, Body mass, pedigree dan age, serta satu atribut akan digunakan sebagai atribut keluaran (output), yaitu Class. Sembilan atrbut masukan merupakan data berjenis scale (skala) yang bernilai angka dan satu atribut keluaran berjenis nominal yaitu klasifikasi nilai 0 (digunakan untuk menyatakan negatif menderita diabetes) dan 1 (digunakan untuk menyatakan positif menderita diabetes).

Sebagian data yang akan diolah terlihat pada gambar 3 dibawah ini.



	gnant	Plasma	bloodpreasure	Tricepskin	seruminsulin	bodymass	pedigree	Age	Class
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
6	5	116	74	0	0	25.6	0.201	30	0
7	3	78	50	32	88	31.0	0.248	26	1
8	10	115	0	0	0	35.3	0.134	29	0
9	2	197	70	45	543	30.5	0.158	53	1
10	8	125	96	0	0	0.0	0.232	54	1
11	4	110	92	0	0	37.6	0.191	30	0
12	10	168	74	0	0	38.0	0.537	34	1
13	10	139	80	0	0	27.1	1.441	57	0
14	1	189	60	23	846	30.1	0.398	59	1
15	5	166	72	19	175	25.8	0.587	51	1
16	7	100	0	0	0	30.0	0.484	32	1
17	0	118	84	47	230	45.8	0.551	31	1

Gambar 3. Data yang akan diolah
Sumber: Penulis (hasil pengolahan data)

Pada gambar 3 di atas terlihat, jika variable Class bernilai 1, maka record tersebut adalah data pasien positif penderita diabetes mellitus, sedangkan jikavariabel Class bernilai 0, maka record tersebut adalah data pasien negative penderita diabetes mellitus

b. Pengolahan Awal Data

1) Pembentukan Sumber Data Random

Ditentukan inisiasi pembangkit aktif (*active generator initialization*), yaitu nilai awal (*starting point*) berupa nilai tetap (*fixed value*) : 9191972. Langkah ini dilakukan agar data yang akan diolah terlebih dahulu diacak agar hasil pengolahan data lebih valid.

2) Pembentukan Variabel Partition

Dari keseluruhan data, sekitar 70 % data ditentukan untuk pembentukan model dan

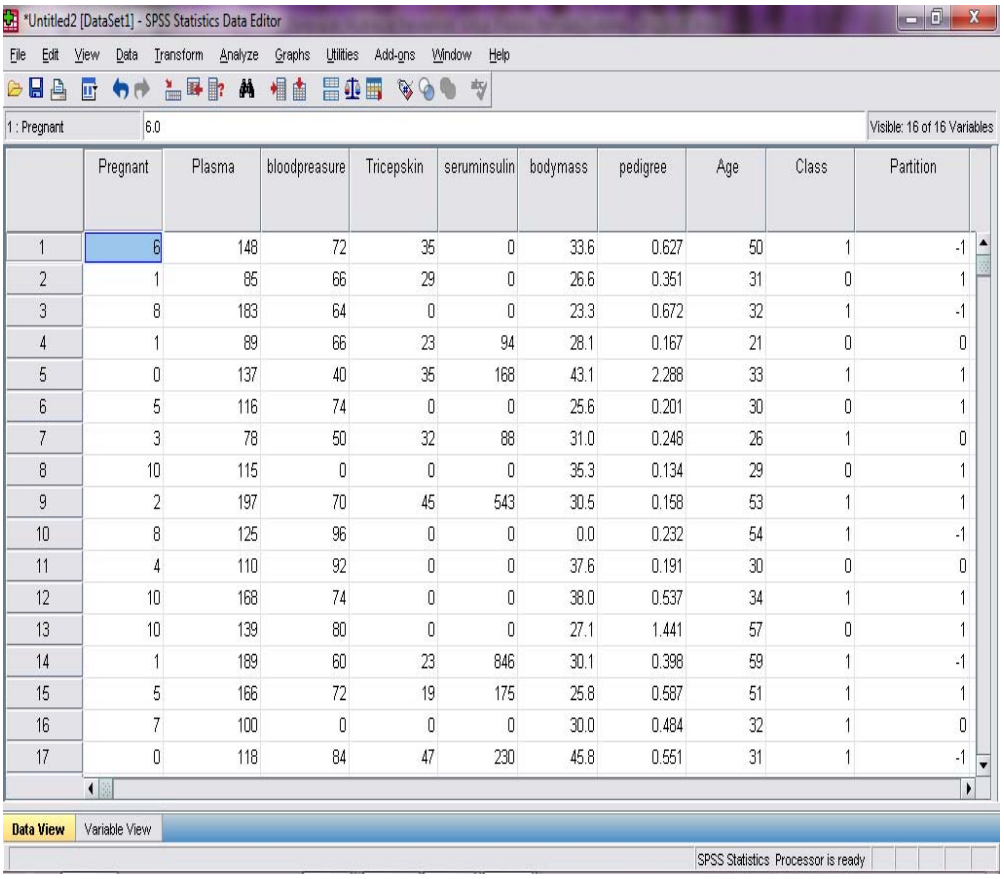
30% ditentukan untuk pengujian model (*holdout sample*) dengan menggunakan *variate Bernoulli* dengan rumus:

2*rv.bernoulli(0.7)-1

Selanjutnya, 70% data untuk pembentukan model diatur kembali menggunakan *variate Bernoulli* dengan rumus:

partition-rv.bernoulli(0.2)

sehingga dari 70% data akan terbagi dua, dimana 80% data digunakan untuk pembentukan awal model (*training sample*) dan 20% untuk memperbaiki bentuk model (*testing sample*).



	Pregnant	Plasma	bloodpreasure	Tricepskin	seruminsulin	bodymass	pedigree	Age	Class	Partition
1	6	148	72	35	0	33.6	0.627	50	1	-1
2	1	85	66	29	0	26.6	0.351	31	0	1
3	8	183	64	0	0	23.3	0.672	32	1	-1
4	1	89	66	23	94	28.1	0.167	21	0	0
5	0	137	40	35	168	43.1	2.288	33	1	1
6	5	116	74	0	0	25.6	0.201	30	0	1
7	3	78	50	32	88	31.0	0.248	26	1	0
8	10	115	0	0	0	35.3	0.134	29	0	1
9	2	197	70	45	543	30.5	0.158	53	1	1
10	8	125	96	0	0	0.0	0.232	54	1	-1
11	4	110	92	0	0	37.6	0.191	30	0	0
12	10	168	74	0	0	38.0	0.537	34	1	1
13	10	139	80	0	0	27.1	1.441	57	0	1
14	1	189	60	23	846	30.1	0.398	59	1	-1
15	5	166	72	19	175	25.8	0.587	51	1	1
16	7	100	0	0	0	30.0	0.484	32	1	0
17	0	118	84	47	230	45.8	0.551	31	1	-1

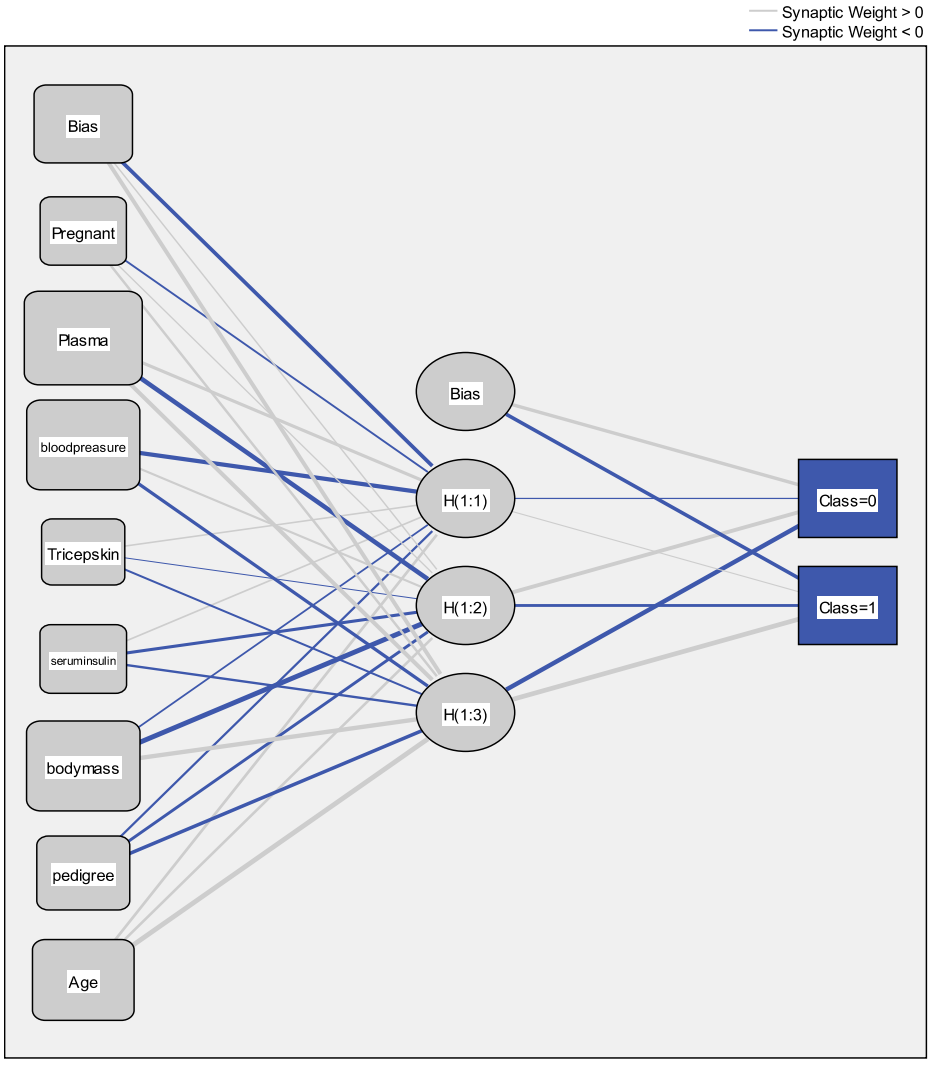
Gambar 4. Data yang telah dipartisi
Sumber: Penulis (hasil pengolahan data)

Pemilahan data menjadi data *training,* data *testing* dan data *holdout* terlihat pada variable partisi (*Partition*). Jika nilai pada variable partisi adalah +1, maka data pada record tersebut dijadian data *training* yaitu digunakan untuk pembentukan model . Jika nilai pada variable partisi adalah -1, maka data pada record tersebut dijadikan data *testing*, yaitu digunakan untuk memperbaiki model. Sedangkan Jika nilai pada variable partisi adalah 0, maka data pada record tersebut dijadian data *holdout*, yaitu digunakan untuk menguji model.

c. Penerapan Algoritma *Neural Network*

1) Struktur *Neural Network*

Data yang telah terbagi menjadi *data training*, *data testing dan data holdout* kemudian diterapkan ke dalam algoritma MLP. Hasil dari penerapan algoritma MLP menghasilkan gambar struktur *neural network* berikut:



Gambar 5. Struktur Neural Network yang dihasilkan
Sumber: Penulis (hasil pengolahan data)

Pada struktur *neural network* yang terbentuk terlihat bahwa struktur terdiri dari:

- Delapan variable *predictor* pada input unit ditambah satu bias
- Tiga *hidden unit* ditambah satu bias
- Dua *output unit*, yaitu Class = 1 untuk positif penderita diabetes mellitus dan Class=0 untuk negative penderita diabetes mellitus

2). Nilai Estimasi Parameter

Berdasarkan struktur neural network tersebut, didapatkan nilai estimasi untuk masing-masing parameter sebagai berikut:

Tabel 3. Nilai Estimasi

		Predicted				
		Hidden Layer 1			Output Layer	
Predictor		H(1:1)	H(1:2)	H(1:3)	[Class=0]	[Class=1]
Input Layer	(Bias)	-.551	.074	.618		
	Pregnant	-.133	.063	.234		
	Plasma	.421	-.765	.639		
	Bloodpreasure	-.716	.184	-.442		
	Tricepskin	.089	-.035	-.166		
	Seruminsulin	.096	-.363	-.227		
	Bodymass	-.099	-1.204	.735		
	Pedigree	-.206	-.326	-.470		
	Age	.295	.237	1.070		
Hidden Layer 1	(Bias)				.450	-.489
	H(1:1)				-.054	.047
	H(1:2)				.603	-.316
	H(1:3)				-.742	.829

Sumber: Penulis (hasil pengolahan data)

Pada tabel 3 terlihat nilai prediksi yang dihasilkan dari masing-masing lapisan tersembunyi (*hidden layer*). Nilai prediksi ini dihasilkan dari pembobotan (*weight*) menggunakan fungsi aktivasi (*activation function*) dari model *neural network* MLP yang digunakan. Nilai hasil perhitungan pembobotan inilah yang nantinya akan menghasilkan nilai prediksi yang diharapkan.

3). Hasil Prediksi

Berdasarkan model yang dihasilkan di atas, maka didapatkan hasil prediksi untuk masing-masing data sebagai berikut:

1 : Pregnant 6.0 — Visible: 16 of 16 Variables

	ass	pedigree	Age	Class	Partition	MLP_PredictedValue	MLP_PseudoProbability_1	MLP_PseudoProbability_2
1	33.6	0.627	50	1	-1	0	0.590	0.410
2	26.6	0.351	31	0	1	0	0.884	0.116
3	23.3	0.672	32	1	-1	1	0.240	0.760
4	28.1	0.167	21	0	0	0	0.987	0.013
5	43.1	2.288	33	1	1	1	0.155	0.845
6	25.6	0.201	30	0	1	0	0.822	0.178
7	31.0	0.248	26	1	0	0	0.924	0.076
8	35.3	0.134	29	0	1	1	0.321	0.679
9	30.5	0.158	53	1	1	1	0.098	0.902
10	0.0	0.232	54	1	-1	0	0.802	0.198
11	37.6	0.191	30	0	0	0	0.665	0.335
12	38.0	0.537	34	1	1	1	0.126	0.874
13	27.1	1.441	57	0	1	0	0.595	0.405
14	30.1	0.398	59	1	-1	1	0.291	0.709
15	25.8	0.587	51	1	1	1	0.239	0.761
16	30.0	0.484	32	1	0	0	0.624	0.376
17	45.8	0.551	31	1	-1	1	0.349	0.651

Gambar 6. Hasil Prediksi masing-masing data
Sumber: Penulis (hasil pengolahan data)

Pada gambar 6 di atas terlihat perbandingan antara variable Class sebagai hasil pemeriksaan terhadap pasien dengan variabel MLP_Prediction_Value sebagai hasil prediksi dari algoritma MLP. Nilai dari MLP_Prediction_value didapatkan berdasarkan perbandingan antara MLP_PseudoProbability_1 dan MLP_PseudoProbability_2. Jika nilai MLP_PseudoProbability_1 lebih besar maka nilai MLP_Prediction_value yang muncul adalah 0. Sedangkan jika nilai MLP_PseudoProbability_2 lebih besar maka nilai MLP_Prediction_value yang muncul adalah 1.

d. Pengukuran Tingkat Akurasi

1) Metode *Confusion Matrix*

Berdasaran hasil penerapan algoritma MLP di atas diukur tingkat akurasinya menggunakan *confusion matrix*. Data diklasifikasikan seperti terlihat pada tabel berikut:

Tabel 4. *Confusion Matrix* MLP

		Predicted		
Sample	Observed	0	1	Percent Correct
Training	0	236	39	85.8%
	1	53	106	66.7%
	Overall Percent	66.6%	33.4%	78.8%
Testing	0	65	11	85.5%
	1	17	20	54.1%
	Overall Percent	72.6%	27.4%	75.2%
Holdout	0	129	19	87.2%
	1	30	42	58.3%
	Overall Percent	72.3%	27.7%	77.7%

Sumber: Penulis (hasil pengolahan data)

Keterangan :
0 : Negatif menderita diabetes mellitus
1 : Positif menderita diabetes mellitus

Pada tabel 4 terlihat hasil evaluasi untuk mengukur tingkat akurasi menggunakan confusion matrix, baik berdasarkan pengolahan *data training*, maupun hasil *data testing* dan *data holdout*. Data tersebut kemudian diklasifikasikan berdasarkan kondisi yang sebenarnya (*sample observe*) dan hasil prediksi (*predicted*). Data tersebut kemudian dihitung tingkat akurasi menggunakan rumus, sehinggadihasilkan nilai prosentase tingkat akurasinya.

Data training digunakan untuk pembentukan model. *Data testing* digunakan untuk memperbaiki model yang dihasilkan dari *data training*. *Data holdout* digunakan untuk mengevaluasi model yang dihasilkan dari *data training* dan *data testing*.. Oleh karenanya, evaluasi yang dilakukan terhadap *data holdout* akan menjadi acuan untuk mengukur tingkat akurasi dari algoritma MLP model *neural network* yang digunakan. Hasil pengukuruan *data holdout* pada tabel 4 adalah:

$$Accuracy = \frac{129 + 42}{129\ 42 + 19 + 30} = \mathbf{0.777273}$$

2) Metode Kurva ROC (*Receiver Operating Characteristic*)

Setelah dievaluasi menggunakan metode *confusion matrix*, penerapan algoritma MLP juga dievaluasi menggunakan kurva ROC untuk mendapatkan nilai AUC (*Area Under the Curve*).

Tabel 4. Nilai AUC (*Area Under the Curve*)

		Area
Class	0	.839
	1	.839

Sumber: Penulis (hasil analisa dari pengolahan data)

Berdasarkan pengelompokan nilai AUC pada *data mining*, maka dapat disimpulkan bahwa hasil prediksi algoritma MLP termasuk dalam **klasifikasi baik (*good classification*)** karena memiliki nilai AUC antara 0.80 sampai 0.90.

5. PENUTUP

akurasinya menggunakan metode pengujian *Confusion Matrix* dan Kurva ROC. Berdasarkan hasil pengukuran tingkat akurasi menggunakan kedua metode tersebut, diketahui bahwa algoritma MLP memiliki tingkat akurasi yang baik. Penelitian ini juga menyimpulkan bahwa hasil pengukuran menggunakan metode *Confusion Matrix* menghasilkan **tingkat akurasi sebesar 0.777273** atau **77,7%** dan menggunakan metode kurva ROC menghasilkan **nilai AUC 0,89** yang termasuk dalam **klasifikasi baik** (good *classification*). Dengan demikian, algoritma MLP dapat digunakan untuk pendeteksian dini penyakit diabete mellitus.

Penelitian ini bertujuan untuk menerapkan algoritma MLP untuk pendeteksian dini penyakit diabetes mellitus. Hasil penerapan algoritma MLP ini kemudian diukur tingkat

DAFTAR PUSTAKA

Bramer, Max. (2007). *Principles of Data Mining*. London: Springer

Departemen Kesehatan Republik Indonesia. (2009). *Profil Kesehatan Indonesia 2008*. Jakarta

Gorunescu, F. (2011). "Data Mining Concepts, Models and Techniques". Berlin Heidelberg: Springer Verlag.

Han, J. & Kember, M. (2006). *Data Mining Concepts and Techniques*. San Fransisco: Morgan Kauffman.

Kusrini, & Luthfi, E. T. (2009). *Algoritma Data Mining.* Yogyakarta: Andi Publishing.

Kothari, C.R. (2004). *Research Methology Methods and Techniques.* India: New Age International Limited.

Kusumadewi, Sri. (2009, Juni 20). *Aplikasi Informatika Medis untuk Penatalaksanaan Diabetes*. Yogyakarta: Seminar Nasional Aplikasi Teknologi Informasi 2009 (SNATI 2009). ISSN : 1907-5022. Pp. C22-C27

Lanywati, Endang. (2001). *Diabetes Mellitus Penyakit Kencing Manis.* Yogyakarta: Kanisius

Larose, D. T. (2005). *Discovering Knowledge in Data.* New Jersey: John Willey & Sons, Inc.

Liao. (2007). *Recent Advances in Data Mining of Enterprise Data: Algorithms and Application*. Singapore: World Scientific Publishing.

Myatt, Glenn J. (2007). *Making Sense of Data: A Practical Guide to Exploratory Data Analysis and Data Mining*. New Jersey: John Wiley & Sons, Inc.

Noertjahyana, Agustinus, & Yulia. (2002, Mei). *Studi Analisa Pelatihan Jaringan Syaraf Tiruan dengan dan tanpa Algoritma Genetika.* Jurnal Informatika. Vol. 3 No. 1. pp13-18.

Obenshain, Mary K. August. (2004). *Application of Data Mining Technique to Healthcare Data.* Statistics for Hospital Epidemiology. Infection Control and Hospital Eidemiology. Vol. 25 No.8.

P. Venkatesan, & S. Anitha. (2006, November 10). *Application of a Radial Basis Function Neural Network for Diagnosis of Diabetes Mellitus.* Current Science, Vol. 91, No. 9. pp. 1195-1199.

Padmavathi P. (2011, January). *A Comparative Sty on Breast Cancer Predition Using RBF and MLP*. International Journal of Scientific &Engineering Research, Volume 2, Issue 1. ISSN 229-5518. pp. 1-5

Ramana, Bendi Venkata, & M. Surendra Prasad Babu. (2012, June). *Liver Classification Using Modified Rotation Forest*. International Journal of Engineering Research and Development, ISSN: 2278-067X, Vol. I, Issue 6, pp. 17-24.

Santoso, Budi. (2007). *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis.* Yogyakarta: Graha Ilmu.

Vercellis, C. (2009). *Business Intelligent: Data Mining and Optimizzation for Decision Making.* Southern Gate, Chichester, West Sussex, United Kingdom : John Wiley & Sons Ltd