KLASIFIKASI UNTUK DIAGNOSA DIABETES MENGGUNAKAN METODE *BAYESIAN REGULARIZATION NEURAL NETWORK (RBNN)*

M.Fadly Rahman[1], M.Ilham Darmawidjadja[2], Dion Alamsah[3]

Teknik Informatika Universitas Padjadjaran
Jl. Raya Bandung Sumedang Km. 21, Jatinangor, Jawa Barat 45363
INDONESIA
email : [1]mfadlyrahman016@gmail.com, [2]milhamd53@gmail.com,
[3]dionduren@gmail.com

Abstrak

Data mining tidak hanya digunakan untuk membahas hal-hal yang berbau informatika, akan tetapi bisa digunakan untuk meneliti berbagai hal, bahkan pola untuk menemukan gejala diabetes pada pasien . Pada paper ini membahas cara memprediksi apakah seseorang mengidap penyakit diabetes menggunakan Data Set yang telah diperoleh dari Machine Learning terlebih dahulu. Pelatihan dilakukan dengan metode Bayesian Regularization Neural Network (RBNN) yang diharapkan dapat memberikan hasil yang diharapkan sesuai dengan prediksi pada penelitian kali ini.

Kata kunci: *Machine Learning, Diabetes,Bayesian Regularization Neural Network, Data Mining, Data Set*

1. PENDAHULUAN

Setiap benda maupun peristiwa pasti memiliki data, salah satunya adalah gejala pada penyakit. Data-data ini sangatlah penting bagi orang-orang yang berkutat di profesi kesehatan untuk menentukan penyakit apa yang sedang diderita oleh pasien yang berobat. Karena kesehatan ini berhubungan dengan nyawa manusia, tentunya kedetilan dan ketelitian sangatlah dibutuhkan agar tidak terjadi kesalahan yang merugikan, bahkan hingga mengambil nyawa manusia.

Salah satu contoh yang diambil di sini adalah gejala dari penyakit Diabetes.Diabetes merupakan salah satu penyakit yang sering dialami oleh banyak orang. Di Indonesia sendiri, jumlah penderita Diabetes Militus (DM) diperkirakan sudah mencapai 10 juta orang, yang membuat Indonesia menempati uruta ke-7 sebagai penderita diabetes terbesar di dunia (Benny Kurniawan, 2016).

Untuk mengklasifikasi gejala diabetes secara cepat dan akurat, tentu diperlukan data -data yang valid dengan metode yang handal agar kesalahan dalam proses klasifikasi dapat diminimalisir. Lalu data tersebut tentu harus diproses agar bisa diterjemahkan menjadi diagnose. Kami memilih mengunakan teknologi Neural Network agar program dapat mempelajari sendiri data-data tersebut dan dapat mengklasifikasi apakah seseorang mengidap diabetes atau tidak.Untuk mengolah data-data tersebut kami menggunakan matlab.

Maka dari itu, harapan kami adalah data yang kami dapatkan serta olah, dapat menjadi salah satu faktor yang membantu dalam hal pendeteksian pengidap penyakit Diabetes sehingga menjadi pengetahuan yang berguna di masa depan.

2. TINJAUAN PUSTAKA

2.1 Data Mining

*Data mining*merupakan disiplin ilmu pengetahuan komputer. Ini adalah proses komputasi menemukan pola dalam data set besar yang melibatkan metode pada kecerdasan buatan, pembelajaran mesin, statistik, dan sistem *database*. Tujuan keseluruhan dari proses data mining adalah untuk mengekstrak informasi dari kumpulan data dan mengubahnya menjadi struktur yang dimengerti untuk digunakan lebih lanjut.

Data mining juga dapat diartikan sebagai kumpulan dari database yang sangat besar yang memiliki ukuran minimal gigabyte yang dapat di ekstrak menjadi sebuah pengetahuan.

Tahap-tahap metodologi *Data Mining*:

2.1.1 *Business Understanding*

Tahap awal ini berfokus pada pemahaman objek projek dan kebutuhan dari perspektif bisnis, dan kemudian mengubah pengetahuan ini ke dalam definisi masalah data mining, dan rencana awal yang dirancang untuk mencapai tujuan.

2.1.2 Data *Understanding*

Fase ini dimulai dengan pengumpulan data dan hasil dengan aktivitas untuk lebih memahami data, mengidentifikasi masalah kualitas data, menemukan wawasan data, atau untuk mendeteksi hal yang menarik untuk membentuk hipotesis untuk informasi yang tersembunyi.

2.1.3 Data *Preparation*

Fase persiapan data mencakup semua kegiatan untuk membangun dataset akhir dari data mentah.Tugasnya dilakukan beberapa kali, dan tidak dalam urutan yang ditentukan.

2.1.4 *Modeling*

Pada fase ini, berbagai teknik pemodelan dipilih dan diterapkan, dan parameternya dikalibrasi menjadi nilai yang optimal. Biasanya, ada beberapa teknik untuk data mining yang memiliki jenis masalah yang sama. Oleh karena itu, kembali ke tahap *Data Preparation* sering dilakukan.

2.1.5 *Evaluation*

Pada tahap ini, projek telah dibangun yang terlihat memiliki kualitas yang tinggi, dari perspektif analisis data. sebelum melakukan penyebaran dari model ini, sangat penting untuk mengevaluasi model, dan meninjau kembali langkah-langkah yang dilakukan untuk membangun model, untuk memastikan bahwa benar-benar mencapai tujuan bisnis.

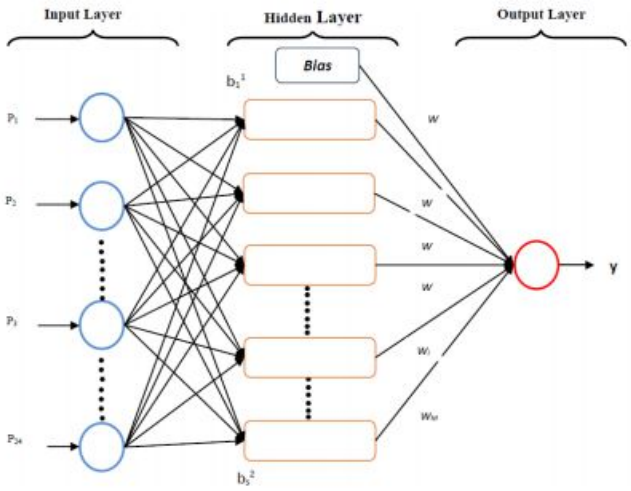
2.1.6 *Deployment*

Penciptaan model umumnya bukan akhir dari proyek. Bahkan jika tujuan dari model ini adalah untuk meningkatkan pengetahuan dari data, pengetahuan yang diperoleh akan perlu diorganisasikan dan disajikan dengan cara yang berguna untuk pelanggan. Jika analis menyebarkan model penting bagi pelanggan untuk memahami depan tindakan yang perlu dilakukan dalam rangka untuk benar-benar memanfaatkan model dibuat.

2.2 *Artificial Neural Network*

*Artificialneuralnetwork(ANN)*terinspirasidarikesadaranatas*complexlearningsystem*pada otakyangterdiridariset-setneuronyangsalingberhubungansecaradekat.Jaringanneuronmampumelakukantugasyan

gsangatkomplekssepertiklasifikasidanpemahamanpola.ANNdapatmemperkirakanrentangyangcukupluassuatumodelstatistikadanfleksibeldalammenggambarkanmodel(liniermaupunnonlinier)[5].ANNdapatdigunakanuntukpermasalahanyangsamadenganpermasalahanstatistikamultivariatseperti*multipleregression*,analisadiskriminan,dananalisakluster.Dalambanyakkasus,hasilyangdidapatdenganANNdapatdibandingkandenganmodel statistikamultivariate.



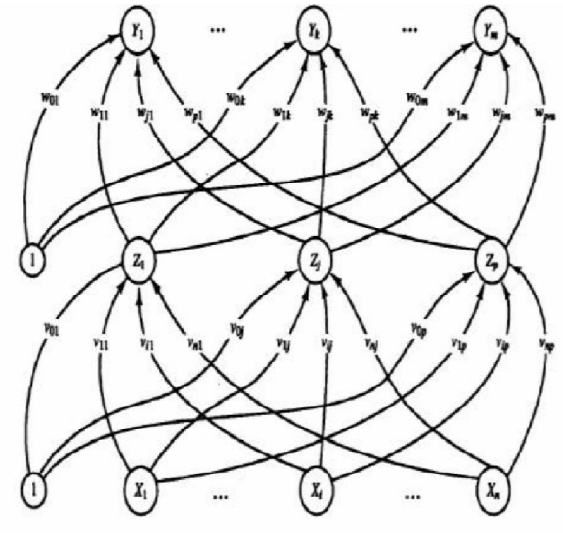
Gambar 1.Arsitektur jaringan pada ANN

Pada arsitektur jaringan ANN, terdapat 3 layer yaitu input layer sebagai masukan nilai yang hendak di training, kemudian nilai masukan tersebut dikirim ke hidden layer dengan cara menghitung nya dengan bobot – bobot yang ada, kemudian dari hidden layer dikirim lagi ke output layer dengan cara yang sama yaitu dengan menghitung bobot – bobot yang ada. Output layer berisikan nilai dari target yang akan menjadi acuan pada tahap pengujian model (testing). Dari hasil pelatihan ini akan didapatkan model *neural network* yang sudah terlatih dimana nilai bobot – bobot nya yang akan menentukan hasil dari prediksi ketika tahap pengujian.

2.3 *Backpropagation*

Pada dasarnya *Neural Network* (NN) merupakan suatu kumpulan elemenelemen pemrosesan sederhana yang saling berhubungan, yang disebut neuron (unit, sel atau node). Setiap neuron dihubungkan dengan neuron lain dengan link komunikasi langsung melalui pola hubungan yang disebut arsitektur jaringan (Fausset, 1994). Tiap-tiap hubungan tersebut mempunyai bobot koneksi (*weight*) yang dilatih untuk mencapai respon yang diinginkan. Sehingga dengan pelatihan terhadap data berdasarkan bobot-bobot koneksi tersebut diharapkan memperoleh output yang diinginkan. Metode yang digunakan untuk menentukan bobot koneksi tersebut dinamakan algoritma pelatihan (*training algorithm*).

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih layar tersembunyi.Gambar 1 adalah arsitektur backpropagation dengan n buah masukan (ditambah sebuah bias), sebuah layar tersembunyi yang terdiri dari p unit (ditambah sebuah bias), serta m buah unit keluaran.



Gambar 2. Arsitektur Backpropagation

Pelatihan pada backpropagation menggunakan metode pancarian titik minimum untuk mencari bobot dengan error minimum. Algoritma backpropagation menggunakan error output untuk mengubah nilai bobot-bobotnya ditahap mundur atau backward (Fausset,1994).Pelatihan backpropagation meliputi 3 fase yaitu fase maju, fase mundur dan fase perubahan bobot.

Fase pertama ialah tahap maju (*feedforward*). Pada tahap ini seluruh proses awal inisialisasi bobot-bobot input dilakukan. Pada tahap ini juga ditentukan angka pembelajaran (α), nilai toleransi error dan jumlah epoch (siklus setiap pola pelatihan) yang diperlukan selama proses komputasi berlangsung. Setelah semua proses inisialisasi dilakukan, maka langkah selanjutnya ialah proses maju. Setiap unit masukan xi akan mengirimkan sinyal masukan ke lapisan tersembunyi. Setelah dihitung dengan menggunakan fungsi aktivasi maka keluarannya akan dikirimkan ke lapisan di atasnya, yaitu lapisan *output*. Setelah nilai keluaran (*yk*) diperoleh, maka dibandingkan dengan target keluaran sebenarnya (*tk*). Selisih *yk – tk* disebut dengan *error* (δk). Jika nilai *error* lebih kecil atau sama dengan dari nilai ambang maka proses terasi dihentikan, tetapi jika tidak maka nilai *error* tersebut digunakan untuk memodifikasi bobot-bobot untuk mengoreksi kesalahan yang terjadi.

Tahap kedua adalah tahap mundur atau backpropagation. Pada tahap ini, nilai *error* (δk) yang diperoleh pada di lapisan *output* digunakan untuk mengoreksi bobot-bobot yang ada pada lapisan tersembunyi yang berhubungan langsung dengan lapisan *output*. Setelah itu nilai *error* (δj) di setiap unit pada lapisan tersembunyi juga dihitung untuk mengoreksi bobot-bobot yang menghubungkan lapisan input dengan lapisan tersembunyi.

Tahap ketiga adalah tahap pengoreksian bobot. Setelah seluruh bobot pada lapisan input dan lapisan tersembunyi dimodifikasi sesuai dengan besar faktor *error*nya, maka ketiga fase ini diulang secara terus menerus sampai kondisi berhenti dipenuhi. Kondisi berhenti yang dimaksud adalah jika jumlah *epoch* yang ditetapkan tercapai atau jika nilai *error* jaringan telah sama dengan atau lebih kecil dari nilai toleransi *error* yang ditetapkan sebelumnya. Pada tahap pelatihan, jaringan diharapkan dapat melatih seluruh data pelatihan yang diberikan untuk mendapatkan bobot akhir jaringan yang akan digunakan pada tahap pengujian.

2.4 *Bayesian Regularization*

Regularisasi berperan meningkatkan proses generalisasi dengan membatasi ukuran bobot suatu jaringan. Jika nilai bobot jaringan lebih kecil maka jaringan akan menanggapi dengan lebih halus. Dengan regularisasi, sebuah jaringan besar yang disederhanakan harus mampu mewakili fungsi yang sebenarnya.

Dalam algoritma Backpropagation klasik bertujuan untuk meminimalkan fungsi $F = Ed$, dimana:

$$E_d = \sum_{i=1}^{n} (t_i - a_i)^2$$

Dalam hal ini n adalah jumlah input pada *training* set, t_i adalah nilai target pada data ke-i dan a_i adalah keluaran untuk data ke-i yang diperoleh sebagai respon jaringan saraf.

Metode regularisasi merubah kinerja kesalahan fungsi dengan menambahkan standar deviasi dari bobot dan bias, yaitu:

$$F = \beta E_d + \alpha E_w$$

$\alpha\beta$, adalah parameter regularisasi, dan E_w didefinisikan sebagai :

$$E_w = \frac{1}{n} \sum_{i=1}^{n} (W_i)^2$$

Wi adalah sebuah bobot atau batas ambang. Dengan menggunakan persamaan untuk mengubah fungsi kinerja error memungkinkan jaringan untuk mendapatkan bobot dan batas ambang terkecil, tetapi tidak bisa menentukan bobot dan batas ambang jaringan yang efektif.

Metode konvensional seringkali sulit untuk menentukan ukuran parameter, Mackay (1992) mengusulkan jaringan yang dapat menyesuaikan ukuran parameter adaptif dengan menggunakan kerangka teori Bayesian, dan memungkinkan tercapainya kinerja yang optimal. Rumus untuk menentukan parameter regularisasi adalah:

$$\begin{cases} \alpha = \dfrac{\gamma}{2E_w} \\ \beta = \dfrac{n - \gamma}{2E_D} \end{cases}$$

Dimana $= n - 2\alpha tr(H)^{-1}$, dimana H adalah matriks *Hessian* dari fungsi F.

2.5 *Confusion Matrix*

Confusion matrix adalah suatu metode yang biasanya digunakan untuk melakukan perhitungan akurasi pada konsep *data mining*.*Confusion matrix* digambarkan dengan tabel yang menyatakan jumlah data uji yang benar diklasifikasikan dan jumlah data uji yang salah diklasifikasikan.

Tabel 1.Tabel *Confusion Matrix*

Correct Classification	*Classified as*	
	Predicted "+"	*Predicted "-"*
Actual "+"	*True Positives*	*False Negatives*
Actual "-"	*False Positives*	*True Negatives*

Berdasarkan tabel *Confusion Matrix* diatas:

a. ***True Positives (TP)*** adalah jumlah *record* datapositif yang diklasifikasikan sebagai nilai positif

b. ***False Positives (FP)*** adalah jumlah *record* data negatif yang diklasifikasikan sebagai nilai positif

c. ***False Negatives (FN)*** adalah jumlah *record* data positif yang diklasifikasikan sebagai nilai positif

d. ***True Negatives (TN)*** adalah jumlah *record* data negatif yang diklasifikasikan sebagai nilai negative

Nilai yang dihasilkan melalui metode *Confusion Matrix* adalah berupa evaluasi sebagai berikut :

a. ***Accuracy,***presentase jumlah *record* data yang diklasifikasikan (prediksi) secara benar oleh algoritma
Rumus : (TP + TN) / Total data = ***Accuracy***

b. ***Misclassification (Error) Rate,*** presentase jumlah *record* data yang diklasifikasikan (prediksi secara salah oleh algoritma.
Rumus : (FP + FN) / Total data = ***Misclassification Rate***

2.6 *Matlab*

Matlab merupakan kepanjangan dari Matrix Laboratory.Struktur data yang terdapat dalam matlab menggunakan matriks atau array berdimensi dua.Oleh Karena itu, penguasaan teori matriks mutlak diperlukan bagi pengguna Matlab agar mudah dalam mempelajari dan memahami operasi yang ada di matlab.Matlab merupakan baasa pemrograman dengan kemampuan tinggi dalam bidang komputasi.Matlab memiliki kemampuan mengintegrasikan komputasi visualisasi dan pemograman.Oleh Karena itu, matlab banyak digunakan dalam bidang riset-riset yang memerlukan komputasi numerik yang kompleks.

Kegunaan umum dari matlab diantaranya untuk matematika dan komputasi, pengembangan algoritma, pemodelan dan simulasi. [1]

3. METODOLOGI PENELITIAN

3.1 *Selection*

Pada tahap ini tertuju pada pemilihan dataset yang akan digunakan. Data yang dipilih harus yang sesuai dengan pertanyaan dari problem yang ada.

3.2 *Pre-Processing*

Pre-processing tidak dilakukan karena kesuluruhan data yang didapatkan sudah normal dan tidak terdapat *missing value*.

3.3 *Transformation*

Karena data sudah siap pakai, kami tidak melakukan tahap transformasi.

3.4 Data *Mining*

Kami menggunakan metode *bayesian regularization neural network* untuk melakukan proses klasifikasi pada data set diabetes yang pada tahap awal adalah melatih atau melakukan proses training pada data – data yang sudah dialokasikan untuk dilatih, hal ini bertujuan untuk membuat model dan menentukan bobot – bobot untuk setiap variasi *datatraining* terhadap target.

3.5 Evaluation

Setelah melakukan proses pelatihan (training), untuk menentukan kualitas atau akurasi dari hasil training sekaligus untuk melakukan testing terhadap model yang telah dibuat, maka dengan menggunakan metode *confusion matrix,* data yang sengaja tidak digunakan untuk training akan dialokasikan untuk menjadi data test untuk melakukan testing terhadap model yang sudah dibuat sekaligus menentukan akurasi dari hasil klasifikasi.

4. HASIL DAN PEMBAHASAN

4.1 Hasil Analisis *Data Set*

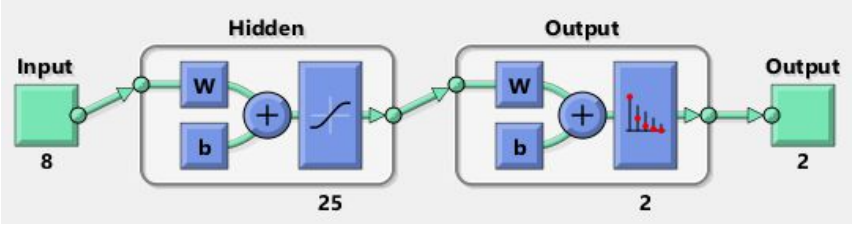
Data set yang digunakan adalah data set *Pima Indians Diabetes* yang diambil dari website UCI, data set ini berisi data – data informasi tentang pasien yang diduga memiliki penyakit diabetes. Keseluruhan data pada data set ini berjumlah 768 *records* yang setiap *record* nya memiliki 9 atribut diantaranya :

a. *Number of times pregnant*
b. *Plasma glucose concentration a 2 hours in an oral glucose tolerance test*
c. *Diastolic blood pressure* (mm Hg)
d. *Triceps skin fold thickness* (mm)
e. *2-Hour serum insulin* (mu U/ml)
f. *Body mass index* (Weight in Kg/(Height in m)^2)
g. *Diabetes pedigree function*
h. *Age* (Years)
i. *Class variable* (0 or 1) ← *target*

Dari atribut data set diatas (1 sampai 8) akan dilakukan proses training & test menggunakan metode *bayesian regularization neural network*, sedangkan atribut ke – 9 akan menjadi target hasil dari proses klasifikasi. dan disini kami akan mencoba menganalisis perbedaan dari akurasi dan error yang didapat dengan melakukan perubahan pada jumlah neuron pada hidden layer.

4.2 Hasil Percobaan & Evaluasi

Pada percobaan ini terdapat 8 atribut yang akan di training dan 2 nilai yang menunjukan target (klasifikasi) pada atribut ke - 9 yang berarti pada *bayesian regularization neural network* di inisialisasi kan 8 *neuron* pada *input layer* , 2 *neuron* pada output layer dengan rasio komposisi data untuk training sebesar 90 % dan data untuk testing sebesar 10 % dari total data di data set.

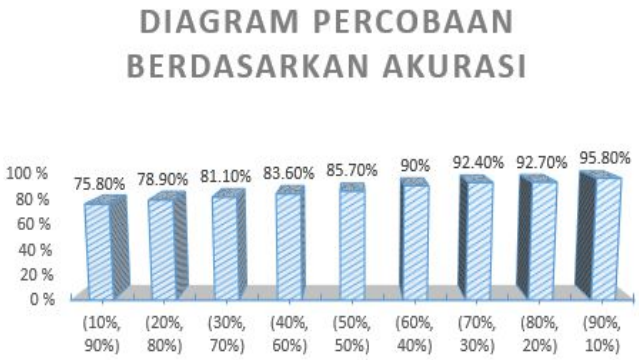


Gambar 3.Arsitektur RBNN yang digunakan

Pada tahap training dilakukan untuk membuat model dan melatih bobot dengan menggunakan metode *bayesian regularization* n*eural network*, dan pada tahap testing model jaringan menggunakan *confusion matrix* untuk menghitung nilai akurasi dan error ketika proses klasifikasi seperti pada tabel dibawah.

Tabel 2.Hasil percobaan

Neuron Hidden Layer	Epoch (Max 1000)	Running Time	Accuracy Rate	Misclassification (Error) Rate
5	96	1 s	81.9 %	18.1 %
10	411	6 s	92.6 %	7.4 %
15	922	21 s	95.4 %	4.6 %
20	1000	30 s	95.3 %	4.7 %
25	889	35 s	96.1 %	3.9 %
30	1000	1 : 33 s	95.5 %	4.5 %



Gambar 4.Diagram Akurasi Berdasarkan Komposisi Training & Testing

Dari sinilah alasan kami memilih rasio komposisi data untuk training sebesar 90 % dan data untuk testing sebesar 10 % dari total data di data set.Karena berdasarkan gambar 3, akurasi tertinggi terdapat pada komposisi (90%, 10%) untuk (training, testing).

Gambar 5.*Evaluasi Confusion Matrix* pada *neuron hidden layer 25*

Dilihat dari Tabel 2 Dan Gambar 4 diatas, dapat diketahui bahwa dengan kita merubah jumlah neuron pada hidden layer, maka akan mempengaruhi nilai akurasi dan error yang didapat menggunakan *Confusion Matrix*. Semakin besar jumlah neuron pada hidden layer maka akan semakin lama *running time* dan akurasi nya pun semakin meningkat (Neuron 5 sampai 25) namun menurun pada neuron ke - 30.
Berdasarkan Gambar 4, maka didapatkan table evaluasi menggunakan *Confusion Matrix* :

Correct Classification	*Classified as*	
	Predicted "+"	*Predicted "-"*
Actual "+"	*255 data (TP)*	*17 data (FN)*
Actual "-"	*13 data (FP)*	*483 data (TN)*

Tabel 3.Hasil evaluasi menggunakan *Confusion Matrix*

Berdasarkan Tabel 3 didapat bahwa terdapat 738 data (TP + TN) dari total 768 data yang diklasifikasikan secara valid (Accurate) dan terdapat 30 data dari total 768 data yang diklasifikasikan secara tidak valid (Error). Dari sini kita dapat menghitung nilai *accuracy rate dan error rate nya* sebagai berikut :

a. **Rumus** : ***Accuracy*** = (TP + TN) / Total data
   = (255 + 483) / 768
   = 738 / 768
   = 0.961 x 100 %
   = 96.1 %
b. **Rumus** : ***Misclassification Rate*** = (FP + FN) / Total data
   = (13 + 17) / 768
   = 30 / 768
   = 0.039 x 100 %
   = 3.9 %.

5. KESIMPULAN

Dari hasil diatas, dapat disimpulkan bahwapada metode *bayesian regularization neural network*, jumlah penggunaan neuron pada hidden layer dapat mempengaruhi akurasi dari hasil proses klasifikasi (semakin banyak maka akan semakin akurat), hal ini dikarenakan dengan kita merubah jumlah neuron pada hidden layer maka kita juga merubah struktur jaringan dari metode *RBNN* ini (Hasil bisa menjadi optimal, bisa juga tidak). Dan menurut hasil percobaan yang kami lakukan, dengan menetapkan model jaringan dengan jumlah neuron pada hidden layer sebesar 25 buah adalah rancangan model jaringan yang paling optimal dalam contoh kasus pada data set diabetes ini.

6. REFERENSI

[1] E. Wahodo, "Matlab," 1 Oktober 2015. [Online]. Available: http://edywahono.web.unej.ac.id/2015/10/01/matlab/. [Diakses Selasa Desember 2016].

[2] A. Indriani, Klasifikasi Data Forum dengan menggunakan Metode Naive Bayes Classifier, 2014.

[3] C. Stergiou dan D. Sigano, "Neural Network," [Online]. Available: https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html. [Diakses Rabu Desember 2016].

[4] A. Hanum, Rabu Agustus 2016. [Online]. Available: http://www.harianjogja.com/baca/2016/08/31/jumlah-penderita-diabetes-indonesia-peringkat-ke-7-di-dunia-749111. [Diakses Rabu Desember 2016].

[5] M. Nielsen, "Neural Networks and Deep Learning," Januari 2016. [Online]. Available: http://neuralnetworksanddeeplearning.com/. [Diakses Rabu Desember 2016].

[6] "Artificial neural network," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network. [Diakses Rabu Desember 2016].

[7] M. H. Meinanda, M. Annisa, N. Muhandri dan K. Suryadi, "Prediksi masa Studi Sarjana dengan Artificial Neural Network," *INTERNETWORKING INDONESIAN JOURNAL,* 2009.