

Optimasi Neural Network Dengan Algoritma Genetika Untuk Prediksi Hasil Pemilukada

Mohammad Badrul ^{1,*}

¹ Program Studi Sistem Informasi; STMIK Nusa Mandiri Jakarta; Jl. Damai No. 8 Warung Jati Barat (Margasatwa) Jakarta Selatan; Telp. (021) 78839513 Fax. (021) 78839421; e-mail : mohammad.mbl@nusamandiri.ac.id

* Korespondensi: e-mail : mohammad.mbl@nusamandiri.ac.id

Diterima: 9 Mei 2016 ; Review: 23 Mei 2016; Disetujui: 1 Juni 2016

Cara citasi: Badrul M. 2016. Optimasi Neural Network Dengan Algoritma Genetika Untuk Prediksi Hasil Pemilukada. Bina Insani ICT Journal. 3 (1): 229 – 242.

Abstrak : Indonesia merupakan salah satu negara demokratis di dunia ini. Negara Indonesia yang terdiri dari beberapa kepulauan melahirkan berbagai macam suku dan budaya. Negara Indonesia yang terdiri dari beberapa kepulauan dibagi menjadi 33 provinsi. Negara Indonesia merupakan Negara demokratis. Pemilu yang diselenggarakan di Indonesia adalah untuk memilih pimpinan baik Presiden dan wakil presiden, anggota DPR, DPRD, dan DPD. Penelitian yang berhubungan dengan pemilu sudah pernah dilakukan oleh peneliti yaitu dengan menggunakan metode decision tree atau dengan menggunakan neural network Dalam penelitian ini dibuatkan model algoritma neural network dan model algoritma neural network berbasis algoritma genetika. Setelah dilakukan pengujian dengan dua model yaitu algoritma neural network dan algoritma genetika maka hasil yang didapat adalah algoritma neural network menghasilkan nilai akurasi sebesar 98,50 % dan nilai AUC sebesar 0,982, namun setelah dilakukan penambahan yaitu algoritma neural network berbasis algoritma genetika nilai akurasi sebesar 93.03 % dan nilai AUC sebesar 0,971.

Kata kunci : algoritma genetika, akurasi, pemilu, neural network

Abstract : Indonesia is one of the democratic countries in the world. State of Indonesia which consists of several islands spawned various tribes and cultures. State of Indonesia which consists of several islands divided into 33 provinces. Indonesia is a democratic country. Elections were held in Indonesia is to choose the heads of both the president and vice president, members of Parliament, Parliament and Council. Research relating to the election had been conducted by researchers is using decision tree method or by using a neural network In this study created a model neural network algorithms and neural network algorithm model based on genetic algorithms. After testing the two models of neural network algorithms and genetic algorithms then the results obtained is a neural network algorithm produces a value of 98.50% accuracy and AUC value of 0.982, but after the addition of a neural network algorithm that is based on a genetic algorithm accuracy value of 93.03 % and AUC value of 0.971.

Keyword: accuracy, elections, genetic algorithm, neural network algorithm

1. Pendahuluan

Indonesia merupakan salah satu negara demokratis di dunia ini. Negara Indonesia yang terdiri dari beberapa kepulauan melahirkan berbagai macam suku dan budaya. Negara Indonesia yang terdiri dari beberapa kepulauan dibagi menjadi 33 provinsi mulai dari Provinsi Nangroe Aceh Darussalam yang ada dibagian paling Barat Negara Indonesia sampai dengan Provinsi Papua Barat yang terletak di Negara Indonesia paling timur. Karena Negara Indonesia merupakan Negara demokratis yang berasaskan pancasila, Pemilihan wakil rakyatpun harus dipilih secara demokratis. Pemilihan umum adalah sarana pelaksanaan kedaulatan rakyat dalam negara

kesatuan Republik Indonesia yang berdasarkan Pancasila dan UUD 1945 (Undang-Undang RI No.10, 2008). Pemilihan umum adalah salah satu pilar utama untuk memilih pemimpin dari sebuah demokrasi atau bisa disebut yang terutama (Santoso, 2004). Pemilu merupakan sarana yang sangat penting karena terselenggaranya sistem politik yang demokratis. Karena itu, tidak mengherankan banyak negara yang ingin disebut sebagai negara demokratis menggunakan pemilu sebagai mekanisme membangun legitimasi. Pemilu bertujuan untuk memilih anggota DPR, DPRD provinsi, dan DPRD kabupaten/kota yang dilaksanakan dengan sistem proporsional terbuka (Undang-Undang RI No.10, 2008). Dengan sistem pemilu langsung dan jumlah partai yang besar maka pemilu legislatif memberikan peluang yang besar pula bagi rakyat Indonesia untuk berkompetisi menaikkan diri menjadi anggota legislatif. Pemilu legislatif tahun 2009 diikuti sebanyak 44 partai yang terdiri dari partai nasional dan partai lokal. Pemilu Legislatif DKI Jakarta Tahun 2009 terdapat 2.268 calon anggota DPRD dari 44 partai yang akan bersaing memperebutkan 94 kursi anggota Dewan Perwakilan Rakyat DKI Jakarta.

Prediksi hasil pemilihan umum perlu diprediksi dengan akurat, karena hasil prediksi yang akurat sangat penting karena mempunyai dampak pada berbagai macam aspek sosial, ekonomi, keamanan, dan lain-lain (Borisjuk, Borisjuk, Rallings, & Thrasher, 2005). Bagi para pelaku ekonomi, peristiwa politik seperti pemilu tidak dapat dipandang sebelah mata, mengingat hal tersebut dapat mengakibatkan risiko positif maupun negatif terhadap kelangsungan usaha yang dijalankan.

Metode prediksi hasil pemilihan umum sudah pernah dilakukan oleh peneliti, (Rigdon, Jacobson, Sewell, & Rigdon, 2009) melakukan prediksi hasil pemilihan umum dengan menggunakan metode *Estimator Bayesian*. (Moscato, Mathieson, Mendes, & Berreta, 2005) melakukan penelitian untuk memprediksi pemilihan presiden Amerika Serikat menggunakan *decision tree*. (Choi & Han, 1999) memprediksi hasil pemilihan presiden di Korea dengan metode *Decision Tree*. (Nagadevara & Vishnuprasad, 2005) memprediksi hasil pemilihan umum dengan model *classification tree* dan *neural network*. (Borisjuk, Borisjuk, Ralling, & Thrasher, 2005) yang memprediksi hasil pemilihan umum dengan menggunakan metode *neural network*.

Decision tree mempunyai kelebihan yaitu mempunyai kelebihan dalam prediksi karena struktur algoritmanya mudah dimengerti dan tingkat kesalahannya cukup kecil sedangkan kelemahan algoritma *decision tree* adalah keandalan cabang yang lebih rendah menjadi lebih buruk dari cabang di atasnya, pohon keputusan yang dihasilkan tidak optimal dan tidak bisa menggunakan sampel yang lebih besar (Sug, 2009), karena itu tidak mudah untuk memahami pohon keputusan besar dan masalah *overfitting* data bisa terjadi dengan target data terbatas yang ditetapkan.

Neural network dapat memecahkan masalah *decision tree* karena memiliki kelebihan pada prediksi *non linear*, memiliki performance yang sangat baik di parallel processing dan kemampuan untuk mentoleransi kesalahan (Xiao & Shao, 2011). Hal ini sangat tepat untuk karakteristik data produksi hasil pemilu pada penelitian ini. Algoritma *neural network* merupakan metode yang sering digunakan untuk memprediksi hasil pemilu legislatif karena data yang disajikan untuk metode ini harus besar dan *non linear* (Gill, 2005). Namun umumnya mempunyai kelemahan pada perlunya data training yang besar dan sulit dalam pemilihan parameter yang optimal (Xiao & Shao, 2011). Perancangan struktur *neural network feedforward* akan mengarah pada meminimalkan dari kesalahan generalisasi, waktu belajar dan dimensi jaringan menyiratkan pembentukan jumlah lapisan, *neuron size* dalam setiap lapisan dan interkoneksi antara *neuron* (Ileana, Rotar, & Incze, 2004). Sementara tidak ada metode yang baik untuk pilihan yang optimal dari dimensi *neural network* itu.

Algoritma genetika telah berkembang sebagai paradigma yang paling banyak digunakan untuk memecahkan masalah optimasi (Alejo, Trevino, & Monorrez, 2008). Algoritma genetika telah terbukti menjadi metode yang efektif untuk digunakan dalam data mining dan pengenalan pola. Sebagian besar aplikasi algoritma genetika dalam pengenalan pola mengoptimalkan beberapa parameter dalam proses klasifikasi, algoritma genetika telah diterapkan untuk menemukan set optimal bobot fitur yang meningkatkan akurasi klasifikasi (Bidgoli & Punch, 2003). Struktur jaringan dengan menggunakan algoritma genetika terbukti efisien dalam memecahkan masalah optimasi dan apalagi banyak teknik evolusi telah dikembangkan untuk menentukan beberapa optima dari fungsi tertentu (Ileana, Rotar, & Incze, 2004). Algoritma genetika merupakan teknik untuk memprediksi kinerja generalisasi berdasarkan sifat statis jaringan seperti *activation function* dan *hidden neuron* akan cukup kuat untuk mencari solusi (Ke

& Liu, 2008). Hal ini dapat memecahkan masalah yang ada pada metode *neural network* yaitu optimasi yang dihasilkan kurang optimal.

Pada penelitian ini algoritma genetika akan diterapkan untuk pemilihan parameter di neural network yaitu *neuron size*, *hidden layer* dan *activation function* yang sesuai dan optimal sehingga hasil prediksi pemilu legislatif DKI Jakarta lebih akurat.

1.1 Pemilihan Umum

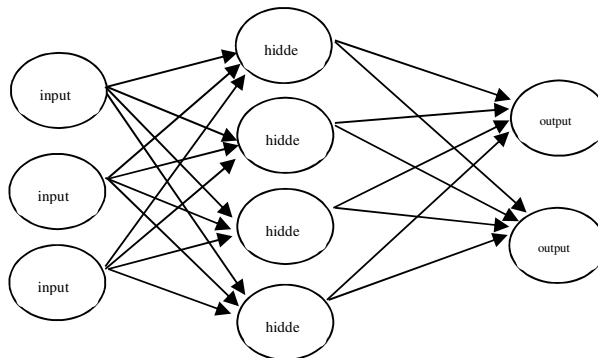
Pemilihan umum adalah salah satu pilar utama dari sebuah demokrasi, kalau tidak dapat yang disebut yang terutama. Pemilu di Indonesia terbagi dari dua bagian, yaitu (Sardini, 2011): yaitu Pemilu orde baru yaitu Sistem pemilihannya dilakukan secara proporsional tidak murni, yang artinya jumlah penentuan kursi tidak ditentukan oleh jumlah penduduk saja tetapi juga didasarkan pada wilayah administrasi dan pemilu era reformasi yaitu dikatakan sebagai pemilu reformasi karena dipercepatnya proses pemilu di tahun 1999 sebelum habis masa kepemimpinan di pemilu tahun 1997. Terjadinya pemilu era reformasi ini karena produk pemilu pada tahun 1997 dianggap pemerintah dan lembaga lainnya tidak dapat dipercaya.

Sistem pemilihan DPR/DPD berdasarkan ketentuan dalam UU nomor 10 tahun 2008 pasal 5 ayat 1 sistem yang digunakan dalam pemilihan legislatif adalah sistem proporsional dengan daftar terbuka, sistem pemilihan DPD dilaksanakan dengan sistem distrik berwakil banyak UU nomor 10 tahun 2008 pasal 5 ayat 2. Menurut UU No. 10 tahun 2008, Peserta pemilihan anggota DPR/D adalah partai politik peserta Pemilu, sedangkan peserta pemilihan anggota DPD adalah perseorangan. Partai politik peserta Pemilu dapat mengajukan calon sebanyak-banyaknya 120 persen dari jumlah kursi yang diperebutkan pada setiap daerah pemilihan demokratis dan terbuka serta dapat mengajukan calon dengan memperhatikan keterwakilan perempuan sekurang-kurangnya 30 %. Partai Politik Peserta Pemilu diharuskan UU untuk mengajukan daftar calon dengan nomor urut (untuk mendapatkan Kursi). Karena itu dari segi pencalonan UU No.10 Tahun 2008 mengadopsi sistem daftar calon tertutup.

UU No.10 Tahun 2008 mengadopsi sistem proporsional dengan daftar terbuka. sistem proporsional merujuk pada formula pembagian kursi dan/atau penentuan calon terpilih, yaitu setiap partai politik peserta pemilu mendapatkan kursi proporsional dengan jumlah suara sah yang diperolehnya. Penerapan formula proporsional dimulai dengan menghitung bilangan pembagi pemilih (BPP), yaitu jumlah keseluruhan suara sah yang diperoleh seluruh partai politik peserta pemilu pada suatu daerah pemilihan dibagi dengan jumlah kursi yang diperebutkan pada daerah pemilihan tersebut.

1.2 Neural Network

Neural network adalah suatu sistem pemroses informasi yang memiliki karakteristik menyerupai dengan jaringan saraf biologi pada manusia. *Neural network* didefinisikan sebagai sistem komputasi di mana arsitektur dan operasi diilhami dari pengetahuan tentang sel saraf biologis di dalam otak, yang merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba menstimulasi proses pembelajaran pada otak manusia tersebut (Astuti, 2009). *Neural network* dibuat berdasarkan model saraf manusia tetapi dengan bagian-bagian yang lebih sederhana. Komponen terkecil dari *neural network* adalah unit atau yang biasa disebut dengan *neuron* dimana *neuron* tersebut akan mentransformasikan informasi yang diterima menuju *neuron* lainnya (Shukla, Tiwari, & Kala, 2010).



Gambar 1. Model *neural network*

Neural network terdiri dari dua atau lebih lapisan, meskipun sebagian besar jaringan terdiri dari tiga lapisan: lapisan input, lapisan tersembunyi, dan lapisan output (Larose, 2005). Pendekatan *neural network* dimotivasi oleh jaringan saraf biologis. Secara kasar, *neural network* adalah satu set terhubung *input/output* unit, di mana masing-masing sambungan memiliki berat yang terkait dengannya. *Neural network* memiliki beberapa properti yang membuat mereka populer untuk *clustering*. Pertama, *neural network* adalah arsitektur pengolahan *inherent* paralel dan terdistribusi. Kedua, *neural network* belajar dengan menyesuaikan bobot interkoneksi dengan data. Hal ini memungkinkan *neural network* untuk "menormalkan" pola dan bertindak sebagai fitur (atribut) *extractors* untuk kelompok yang berbeda. Ketiga, *neural network* memproses vektor numerik dan membutuhkan pola objek untuk diwakili oleh fitur kuantitatif saja (Gorunescu, 2011).

Neural network terdiri dari kumpulan node (*neuron*) dan relasi. Ada tiga tipe node (*neuron*) yaitu, *input*, *hidden* dan *output*. Setiap relasi menghubungkan dua buah node dengan bobot tertentu dan juga terdapat arah yang menunjukkan aliran data dalam proses (Kusrini & Luthfi, 2009). Kemampuan otak manusia seperti mengingat, menghitung, menggeneralisasi, adaptasi, diharapkan *neural network* dapat meniru kemampuan otak manusia. *Neural network* berusaha meniru struktur/arsitektur dan cara kerja otak manusia sehingga diharapkan bisa dan mampu menggantikan beberapa pekerjaan manusia. *Neural network* berguna untuk memecahkan persoalan yang berkaitan dengan pengenalan pola, klasifikasi, prediksi dan data mining (Shukla, Tiwari, & Kala, 2010).

Input node terdapat pada *layer* pertama dalam *neural network*. Secara umum setiap input node merepresentasikan sebuah input parameter seperti umur, jenis kelamin, atau pendapatan. *Hidden node* merupakan node yang terdapat di bagian tengah. *Hidden node* ini menerima masukan dari input node pada *layer* pertama atau dari *hidden node* dari *layer* sebelumnya. *Hidden node* menggabungkan semua masukan berdasarkan bobot dari relasi yang terhubung, mengkalkulasikan, dan memberikan keluaran untuk *layer* berikutnya. *Output node* mempresentasikan atribut yang diprediksi (Kusrini & Luthfi, 2009).

Setiap node (*neuron*) dalam *neural network* merupakan sebuah unit pemrosesan. Tiap node memiliki beberapa masukan dan sebuah keluaran. Setiap node mengkombinasikan beberapa nilai masukan, melakukan kalkulasi, dan membangkitkan nilai keluaran (aktifasi). Dalam setiap node 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 (Kusrini & Luthfi, 2009). Serta beberapa fungsi aktifasi yang dapat digunakan yaitu *heaviside (threshold)*, *step activation*, *piecewise*, *linear*, *gaussian*, *sigmoid*, *hyperbolic tangent* (Gorunescu, 2011).

Salah satu keuntungan menggunakan *neural network* adalah bahwa *neural network* cukup kuat sehubungan dengan data. Karena *neural network* berisi banyak node (*neuron* buatan) dengan bobot ditugaskan untuk setiap koneksi (Larose, 2005).

Aplikasi *neural network* telah banyak dimanfaatkan untuk berbagai kepentingan seperti di bidang Elektronik, Otomotif, Perbankan, Sistem penerbangan udara, Dunia hiburan, transportasi publik, telekomunikasi, bidang Kesehatan, Keamanan, bidang Robotika, Asuransi, Pabrik, Financial, Suara, Pertambangan dan sistem pertahanan (Astuti, 2009). Metode pelatihan terbimbing/terawasi (*supervised learning*) adalah pelatihan yang memasukkan target keluaran dalam data untuk proses pelatihan. Ada beberapa metode pelatihan terbimbing yang telah ditemukan oleh para peneliti, diantaranya adalah yang paling sering diaplikasikan adalah *perceptron* dan *backpropagation (feedforward)*.

Algoritma yang paling populer pada algoritma *neural network* adalah algoritma *backpropagation*. Algoritma pelatihan *backpropagation* atau ada yang menterjemahkan menjadi propagasi balik pertama kali dirumuskan oleh Paul Werbos pada tahun 1974 dan dipopulerkan oleh Rumelhart bersama McClelland untuk dipakai pada *neural network*. Metode *backpropagation* pada awalnya dirancang untuk *neural network feedforward*, tetapi pada perkembangannya, metode ini diadaptasi untuk pembelajaran pada model *neural network* lainnya (Astuti, 2009). Salah satu metode pelatihan terawasi pada *neural network* adalah metode *backpropagation*, di mana ciri dari metode ini adalah meminimalkan *error* pada output yang dihasilkan oleh jaringan.

Algoritma *backpropagation* mempunyai pengatur hubungan yang sangat sederhana yaitu: jika keluaran memberikan hasil yang salah, maka penimbang (*weight*) dikoreksi supaya galatnya dapat diperkecil dan respon jaringan selanjutnya diharapkan akan mendekati nilai yang

benar. Algoritma ini juga berkemampuan untuk memperbaiki penimbang pada lapisan tersembunyi (*hidden layer*) (Purnomo & Kurniawan, 2006).

Inisialisasi awal bobot jaringan *backpropagation* yang terdiri atas lapisan input, lapisan tersembunyi, dan lapisan output (Astuti, 2009). Tahap pelatihan *backpropagation* merupakan langkah untuk melatih suatu *neural network* yaitu dengan cara melakukan perubahan penimbang (sambungan antar lapis yang membentuk *neural network* melalui masing-masing unitnya). Sedangkan penyelesaian masalah akan dilakukan jika proses pelatihan tersebut telah selesai, fase ini disebut dengan fase *mapping* atau proses pengujian/testing.

Berikut 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:

$$\text{Input } j = \sum_{i=1}^n o_i w_{ij} + \theta_j$$

3. Berdasarkan input dari langkah dua, selanjutnya membangkitkan output. Untuk simpul menggunakan fungsi aktivasi sigmoid:

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

4. Hitung nilai *Error* antara nilai yang diprediksi dengan nilai yang sesungguhnya menggunakan rumus: $\text{Error}_j = \text{output}_j \cdot (1 - \text{output}_j) \cdot (\text{Target}_j - \text{Output}_j)$
5. Setelah nilai *Error* dihitung, selanjutnya dibalik ke *layer* sebelumnya (*backpropagation*). Untuk
6. menghitung nilai *Error* pada *hidden layer*, menggunakan rumus:

$$\text{Error}_j = \text{Output}_j(1 - \text{Output}_j) \sum_{k=1}^n \text{Error}_k W_{jk}$$

7. Nilai *Error* yang dihasilkan dari langkah sebelumnya digunakan untuk memperbarui bobot relasi menggunakan rumus:

$$W_{ij} = W_{ij} + \eta \cdot \text{Error}_j \cdot \text{Output}_i$$

1.3 Algoritma Genetika

Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi dari kromosom antar individu organisme. Variasi kromosom ini akan mempengaruhi laju reproduksi dan tingkat kemampuan organisme untuk tetap hidup (Kusumadewi & Purnomo, 2005). Pada dasarnya ada 4 kondisi yang sangat mempengaruhi proses evaluasi, yakni sebagai berikut :

- a. Kemampuan organisme untuk melakukan reproduksi
- b. Keberadaan populasi organisme yang bisa melakukan reproduksi
- c. Keberadaan organisme dalam suatu populasi
- d. Perbedaan kemampuan untuk survive.

Individu yang lebih kuat(fit) akan memiliki tingkat survival dan tingkat reproduksi yang lebih tinggi jika dibandingkan dengan individu yang kurang fit. Pada kurun waktu tertentu (sering dikenal dengan istilah generasi), populasi secara keseluruhan akan lebih banyak memuat organisme yang fit (Kusumadewi & Purnomo, 2005).

Algoritma genetika pertama kali dirintis John Holland dari Universitas Michigan pada tahun 1960-an, algoritma genetika telah diaplikasikan secara luas pada berbagai bidang. algoritma genetika banyak digunakan untuk memecahkan masalah optimasi, walaupun pada kenyataannya juga memiliki kemampuan yang baik untuk masalah-masalah selain optimasi. John Holland menyatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika. algoritma genetika adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom (Haupt & Haupt, 2004).

Pada algoritma genetika, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan istilah generasi. Pada setiap generasi,

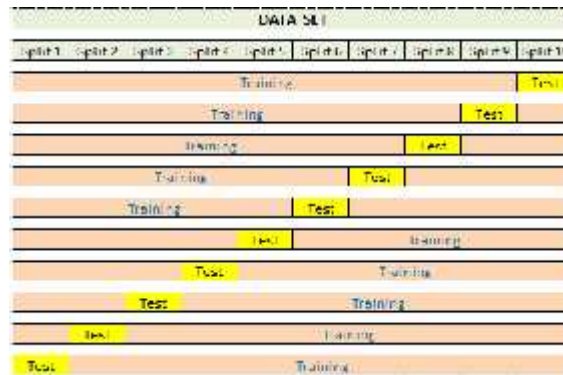
kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi fitness. Nilai fitness dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah anak (*offspring*) terbentuk dari gabungan 2 kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyalangan (*crossover*). Selain operator penyalangan, suatu kromosom dapat juga dimodifikasi dengan menggunakan operator mutasi. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai fitness dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melakukan berbagai generasi, maka algoritma ini akan konvergen ke kromosom yang terbaik (Kusumadewi & Purnomo, 2005).

Misalkan P (generasi) adalah populasi dari suatu generasi, maka secara sederhana algoritma genetika terdiri dari langkah-langkah berikut (Larose D. T., 2006):

- a. Langkah 0 : inisialisasi
- b. Asumsikan bahwa data yang dikodekan dalam string bit (1 dan 0). Tentukan probabilitas crossover atau p_c Crossover rate dan probabilitas mutasi atau laju mutasi p_m . Biasanya, p_c dipilih menjadi cukup tinggi (misalnya, 0,7), dan p_m dipilih sangat rendah (misalnya, 0,001)
- c. Langkah 1: Populasi yang dipilih, terdiri dari satu set n kromosom setiap panjang i.
- d. Langkah 2: cocokkan $f(x)$ dihitung untuk setiap kromosom dalam populasi.
- e. Langkah 3: ulangi melalui langkah-langkah berikut sampai n keturunan telah dihasilkan
 - 1) Langkah 3a: Seleksi. Menggunakan nilai-nilai dari fungsi fitness $f(x)$ dari langkah 2, menetapkan probabilitas seleksi untuk setiap kromosom dengan fitness yang lebih tinggi memberikan probabilitas yang lebih tinggi seleksi. Istilah yang biasa untuk cara probabilitas ini ditugaskan adalah metode roda roulette. Untuk setiap kromosom x_i , menemukan proporsi kebugaran ini kromosom untuk total kebugaran menyimpulkan atas semua kromosom. Artinya, menemukan $f(x_i) / \sum f(x_i)$ dan menetapkan proporsi ini menjadi probabilitas memilih bahwa kromosom untuk menjadi orang tua. Kemudian pilih sepasang kromosom untuk menjadi orangtua, berdasarkan probabilitas. Biarkan kromosom yang sama memiliki potensi yang akan dipilih untuk menjadi orangtua yang lebih dari sekali. Membiarkan kromosom untuk berpasangan dengan dirinya sendiri akan menghasilkan salinan pohon kromosom yang ke generasi baru. Jika analisis prihatin konvergen ke optimum lokal terlalu cepat, mungkin pasangan tersebut seharusnya tidak diperbolehkan.
 - 2) Langkah 3b: Crossover. Pilih lokus yang dipilih secara acak (titik crossover) untuk tempat melakukan crossover. Kemudian, dengan p_c probabilitas, melakukan crossover dengan orang tua yang dipilih dalam langkah 3a: sehingga membentuk dua keturunan baru. Jika crossover tidak dilakukan, mengkopi dua salinan tepat dari orang tua untuk menjadi diteruskan kepada generasi baru.
 - 3) Langkah 3c: Mutasi. Dengan p_m probabilitas, melakukan mutasi pada masing-masing keturunan dua pada setiap titik lokus. Kromosom kemudian mengambil tempat mereka di populasi baru. Jika n ganjil, buang satu kromosom baru secara acak.
- a. Langkah 4 : populasi kromosom baru menggantikan populasi saat ini
- b. Langkah 5: Periksa apakah kriteria penghentian telah dipenuhi. Sebagai contoh, adalah perubahan kebugaran rata-rata dari generasi ke generasi makin kecil? Jika konvergensi tercapai, berhenti dan melaporkan hasil, jika tidak, lanjutkan ke langkah 2.

1.4. Pengujian K-Fold Cross Validation

Cross Validation adalah teknik validasi dengan membagi data secara acak kedalam k bagian dan masing-masing bagian akan dilakukan proses klasifikasi (Han dan Kamber, 2007). Dengan menggunakan *cross validation* akan dilakukan percobaan sebanyak k. Data yang digunakan dalam percobaan ini adalah data *training* untuk mencari nilai *error rate* secara keseluruhan. Secara umum pengujian nilai k dilakukan sebanyak 10 kali untuk memperkirakan akurasi estimasi. Dalam penelitian ini nilai k yang digunakan berjumlah 10 atau *10-fold Cross Validation*.



Sumber : Han dan Kamber, (2007)

Gambar 4. Ilustrasi 10 Fold Cross Validation

Pada gambar 4 terlihat bahwa tiap percobaan akan menggunakan satu data *testing* dan k-1 bagian akan menjadi data *training*, kemudian data *testing* itu akan ditukar dengan satu buah data *training* sehingga untuk tiap percobaan akan didapatkan data *testing* yang berbeda-beda.

1.5 Confusion Matrix

Confusion matrix memberikan keputusan yang diperoleh dalam *training* dan *testing*, *confusion matrix* memberikan penilaian *performance* klasifikasi berdasarkan objek dengan benar atau salah (Gorunescu, 2011).

Confusion matrix berisi informasi aktual (*actual*) dan prediksi (*predicted*) pada sistem klasifikasi. Berikut tabel penjelasan tentang confusion matrix.

Tabel 1. Confusion Matrix

Classification	Predicted Class		
		Class = Yes	Class = No
	Observed Class		
	Class = Yes	A (True Positif-tp)	B (False negatif-fn)
	Class = No	C (False positif- fp)	D (true negative-tn)

Sumber : Gorunescu, (2011)

Keterangan:

True Positive (tp) = proporsi positif dalam data set yang diklasifikasikan positif

True Negative (tn) = proporsi negative dalam data set yang diklasifikasikan negative

False Positive (fp) = proporsi negatif dalam data set yang diklasifikasikan positif

FalseNegative(fn) = proporsi negative dalam data set yang diklasifikasikan negatif

Berikut adalah persamaan model *confusion matrix*:

- Nilai akurasi (acc) adalah proporsi jumlah prediksi yang benar.

Dapat dihitung dengan menggunakan persamaan:

$$acc = \frac{tp + tn}{tp + tn + fp + fn}$$

- Sensitivity digunakan untuk membandingkan proporsi tp terhadap tupel yang positif, yang dihitung dengan menggunakan persamaan:

$$Sensitivity = \frac{tp}{tp + fn}$$

- Specificity digunakan untuk membandingkan proporsi tn terhadap tupel yang negatif, yang dihitung dengan menggunakan persamaan:

$$\text{Specificity} = \frac{tn}{tn + fp}$$

d. PPV (*positive predictive value*) adalah proporsi kasus dengan hasil diagnosa positif, yang dihitung dengan menggunakan persamaan:

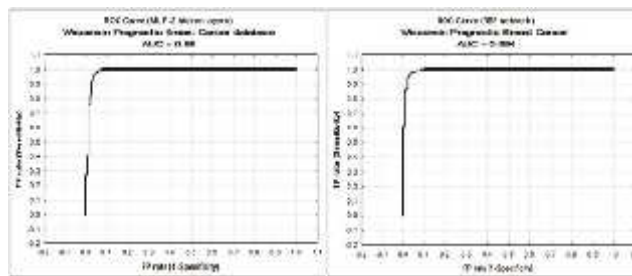
$$PPV = \frac{tp}{tp + fp}$$

e. NPV (*negative predictive value*) adalah proporsi kasus dengan hasil diagnosa negatif, yang dihitung dengan menggunakan persamaan:

$$NPV = \frac{tn}{tn + fn}$$

1.6 Curve ROC

Curve ROC (*Receiver Operating Characteristic*) adalah cara lain untuk mengevaluasi akurasi dari klasifikasi secara visual (Vercellis, 2009). Sebuah grafik ROC adalah plot dua dimensi dengan proporsi positif salah (fp) pada sumbu X dan proporsi positif benar (tp) pada sumbu Y. Titik (0,1) merupakan klasifikasi yang sempurna terhadap semua kasus positif dan kasus negatif. Nilai positif salah adalah tidak ada (fp = 0) dan nilai positif benar adalah tinggi (tp = 1). Titik (0,0) adalah klasifikasi yang memprediksi setiap kasus menjadi negatif {-1}, dan titik (1,1) adalah klasifikasi yang memprediksi setiap kasus menjadi positif {1}. Grafik ROC menggambarkan *trade-off* antara manfaat ('*true positives*') dan biaya ('*false positives*'). Berikut tampilan dua jenis kurva ROC (*discrete* dan *continous*).



Sumber : Gorunescu, (2011)

Gambar 5. Grafik ROC (*discrete* dan *continous*)

Pada Gambar 2.4 garis diagonal membagi ruang ROC, yaitu:

1. (a) poin diatas garis diagonal merupakan hasil klasifikasi yang baik.
2. (b) point dibawah garis diagonal merupakan hasil klasifikasi yang buruk.

Dapat disimpulkan bahwa, satu point pada kurva ROC adalah lebih baik dari pada yang lainnya jika arah garis melintang dari kiri bawah ke kanan atas didalam grafik. Tingkat akurasi dapat di diagnosa sebagai berikut (Gorunescu, 2011):

Akurasi 0.90 – 1.00 = *Excellent classification*

Akurasi 0.80 – 0.90 = *Good classification*

Akurasi 0.70 – 0.80 = *Fair classification*

Akurasi 0.60 – 0.70 = *Poor classification*

Akurasi 0.50 – 0.60 = *Failure*

2. Metode Penelitian

Menurut Sharp et al (Dawson, 2009) penelitian adalah mencari melalui proses yang metodis untuk menambahkan pengetahuan itu sendiri dan dengan yang lainnya, oleh penemuan fakta dan wawasan tidak biasa. Pengertian lainnya, penelitian adalah sebuah kegiatan yang bertujuan untuk membuat kontribusi orisinal terhadap ilmu pengetahuan (Dawson, 2009).

Penelitian ini adalah penelitian eksperimen dengan metode penelitian sebagai berikut

1. Pengumpulan data

Pada bagian ini dijelaskan tentang bagaimana dan darimana data dalam penelitian ini didapatkan, ada dua tipe dalam pengumpulan data, yaitu pengumpulan data primer dan pengumpulan data sekunder. Data primer adalah data yang dikumpulkan pertama kali untuk

melihat apa yang sesungguhnya terjadi. Data sekunder adalah data yang sebelumnya pernah dibuat oleh seseorang baik di terbitkan atau tidak (Kothari, 2004). Pengumpulan data primer bisa didapat dari model interview terhadap responden, maupun dengan model observasi terhadap suatu badan yang sedang diteliti (Gray, 2004). Pada tahap ini ditentukan data yang akan diproses. Mencari data yang tersedia, memperoleh data tambahan yang dibutuhkan, mengintegrasikan semua data kedalam data set, termasuk variabel yang diperlukan dalam proses.

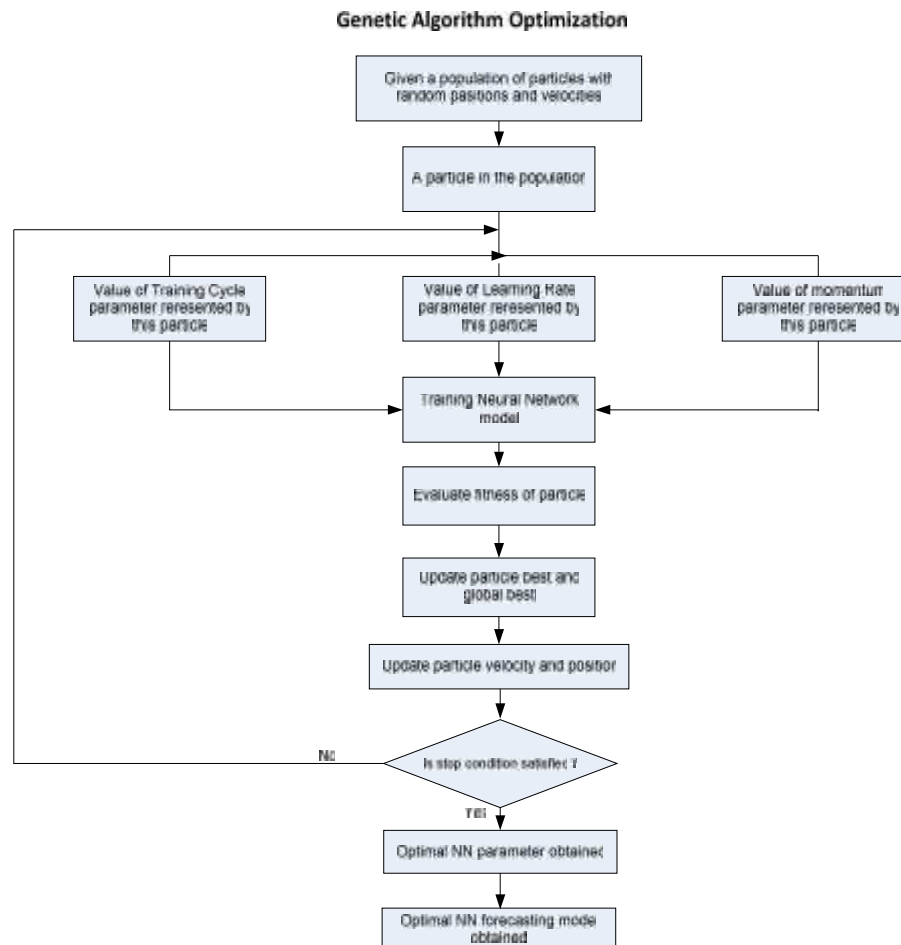
2. Pengolahan awal data

Pada bagian ini dijelaskan tentang tahap awal data mining. Pengolahan awal data meliputi proses input data ke format yang dibutuhkan, pengelompokan dan penentuan atribut data, serta pemecahan data (split) untuk digunakan dalam proses pembelajaran (training) dan pengujian (testing).

3. Model yang diusulkan

Pada tahap ini data dianalisis, dikelompokkan variabel mana yang berhubungan dengan satu sama lainnya. Setelah data dianalisis lalu diterapkan model-model yang sesuai dengan jenis data. Pembagian data kedalam data latihan (*training data*) dan data uji (*testing data*) juga diperlukan untuk pembuatan model.

Model yang diusulkan pada penelitian ini berdasarkan *state of the art* tentang prediksi hasil pemilihan umum adalah dengan menerapkan *neural network* dan *neural network* berbasis algoritma genetika, yang terlihat pada Gambar dibawah ini



Gambar 2. Model yang diusulkan

Pada Gambar 2 menunjukkan proses yang dilakukan dalam tahap modeling untuk menyelesaikan prediksi hasil pemilihan umum dengan menggunakan dua metode yaitu algoritma *neural network* dan algoritma *neural network* dengan algoritma genetika.

4. Eksperiment dan pengujian model

Untuk memilih arsitektur *neural network* yang tepat, agar menghasilkan *nilai akurasi dan nilai AUC yang terbesar*, diperlukan pengaturan (*adjustment*) untuk parameter-parameter *neural network* antara lain *training cycle*, *learning rate*, *momentu*, *hidden layer* dan *neuron size*. Untuk memilih arsitektur *neural network* yang tepat, agar menghasilkan *nilai akurasi dan nilai AUC yang terbesar*, diperlukan pengaturan (*adjustment*) untuk parameter-parameter *neural network*. Berikut ini adalah parameter-parameter yang membutuhkan *adjustment*:

a. *Training cycle*, *learning rate*, dan *momentum*

Training cycle adalah jumlah perulangan training yang perlu dilakukan untuk mendapatkan error yang terkecil. Nilai *training cycle* bervariasi mulai dari 1 sampai dengan tak terhingga. *Learning rate* adalah variabel yang digunakan oleh algoritma pembelajaran untuk menentukan bobot dari *neuron* (K & Deepa, 2011). Nilai yang besar menyebabkan pembelajaran lebih cepat tetapi ada osilasi bobot, sedangkan nilai yang kecil menyebabkan pembelajaran lebih lambat. Nilai *learning rate* harus berupa angka positif kurang dari 1. *Momentum* digunakan untuk meningkatkan *convergence*, mempercepat waktu pembelajaran dan mengurangi osilasi. Nilai *momentum* bervariasi dari 0 ke 1.

b. *Hidden Layer*

Ada 2 masalah dalam pengaturan *hidden layer*, yaitu penentuan jumlah *hidden layer* dan penentuan *size* atau jumlah *neuron* dari *hidden layer*. Saat ini tidak ada alasan teoritis untuk menggunakan *neural network* dengan lebih dari dua *hidden layer*. Bahkan, untuk banyak masalah praktis, tidak ada alasan untuk menggunakan lebih dari satu *hidden layer* (K & Deepa, 2011).

Penentuan jumlah *neuron* yang terlalu sedikit akan mengakibatkan *underfitting*, yaitu jaringan kurang dapat mendeteksi sinyal atau pola dalam set data. Jumlah *neuron* yang terlalu banyak akan mengakibatkan *overfitting*, yaitu jumlah informasi dalam *training set* yang terbatas, tidak cukup untuk melatih semua *neuron* dalam *hidden layer*.

c. Arsitektur *neural network*

Arsitektur *neural network* tersusun dari tiga buah lapisan (*layer*), yaitu *input*, *hidden layer*, dan *output*. *hidden layer* terletak diantara input dan output.

5. Evaluasi dan validasi hasil

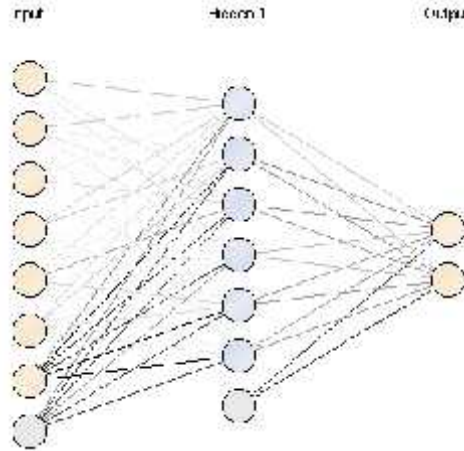
Setelah ditemukan nilai akurasi yang paling ideal dari parameter di atas akan terbentuk struktur algoritma yang ideal untuk pemecahan masalah tersebut. Model yang diusulkan pada penelitian tentang prediksi hasil pemilihan umum adalah dengan menerapkan *neural network* dan *neural network* dengan optimasi algoritma genetika. Penerapan algoritma *neural network* dengan menentukan nilai *training cycle* terlebih dahulu. Setelah didapatkan nilai akurasi dan AUC terbesar, nilai *training cycle* tersebut akan dijadikan nilai yang akan digunakan untuk mencari nilai akurasi dan AUC tertinggi pada *learning rate* dan *momentum*. Setelah ditemukan nilai yang paling tinggi dari *training cycle*, *learning rate* dan *momentum* selanjutnya adalah menentukan ukuran (*size*) pada *hidden layer* tersebut. Sedangkan penerapan algoritma *neural network* dengan optimasi algoritma genetika berdasarkan pada nilai *training cycle* pada algoritma tersebut. Setelah ditemukan nilai akurasi yang paling ideal dari parameter tersebut langkah selanjutnya adalah menentukan nilai *learning rate* dan *momentum*. Setelah ditemukan nilai akurasi yang paling ideal dari parameter tersebut langkah selanjutnya adalah menentukan *hidden layer* dan *neuron size* sehingga terbentuk struktur algoritma yang ideal untuk pemecahan masalah tersebut.

3. Hasil dan Analisis

3.1 Metode Neural Network

Algoritma *neural network* adalah algoritma untuk pelatihan *supervised* dan didesain untuk operasi pada *feed forward* multilapis. Algoritma *neural network* bisa didekripsikan sebagai berikut: ketika jaringan diberikan pola masukan sebagai pola pelatihan maka pola tersebut menuju ke unit-unit pada lapisan tersembunyi untuk diteruskan ke unit-unit lapisan terluar. Hasil terbaik pada *eksperiment* adalah dengan *accuracy* yang dihasilkan sebesar 98.50 dan AUCnya 0.982.

Dari eksperimen terbaik di atas maka didapat arsitektur *neural network* dengan menghasilkan enam *hidden layer* dengan tujuh atribut *input layer* dan dua *output layer*. Gambar arsitektur *neural network* terlihat pada gambar 3 seperti di bawah ini



Gambar 3. Arsitektur *neural network*

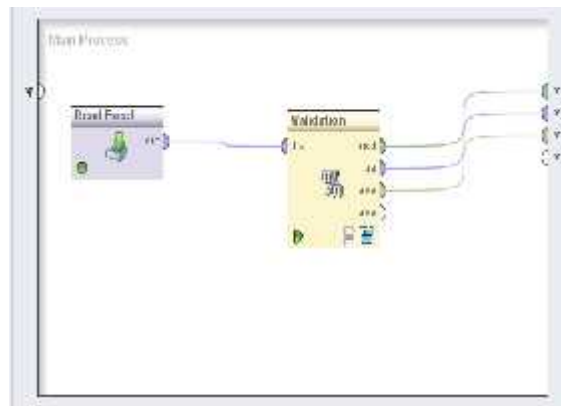
3.2 Neural Network berbasis Algoritma Genetika

Algoritma *neural network* bisa dideksripsikan sebagai berikut: ketika jaringan diberikan pola masukan sebagai pola pelatihan maka pola tersebut menuju ke unit-unit pada lapisan tersembunyi untuk diteruskan ke unit-unit lapisan terluar. Sedangkan Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi dari kromosom antar individu organisme. Variasi kromosom ini akan mempengaruhi laju reproduksi dan tingkat kemampuan organisme untuk tetap hidup. Pengujian dengan menggunakan *neural network* berbasis *algoritma genetika* didapatkan nilai *accuracy* 93.03 % dengan nilai *precision* 91.28 % dan nilai *AUC* adalah 0.971.

3.3 Analisa Evaluasi dan Validasi Model

Hasil dari pengujian model yang dilakukan adalah memprediksi hasil pemilu legislatif DKI Jakarta 2009 dengan *neural network* dan *neural network* berbasis algoritma genetika untuk menentukan nilai *accuracy* dan *AUC*.

Dalam menentukan nilai tingkat keakurasian dalam model *neural network* dan algoritma *neural network* berbasis algoritma genetika. Metode pengujiannya menggunakan *cross validation* dengan desain modelnya sebagai berikut.



Gambar 5. Pengujian cross validation

Dari hasil pengujian diatas, baik evaluasi menggunakan *counfusion matrix* maupun ROC *curve* terbukti bahwa hasil pengujian algoritma *neural network* berbasis algoritma genetika memiliki nilai akurasi yang lebih tinggi dibandingkan dengan algoritma *neural network*.

Nilai akurasi untuk model algoritma *neural network* sebesar 98.50 % dan nilai akurasi untuk model algoritma *neural network* berbasis Algoritma genetika sebesar 93.03 % dengan selisih akurasi 5.47 %.

Sedangkan evaluasi menggunakan ROC *curve* sehingga menghasilkan nilai AUC (*Area Under Curve*) untuk model algoritma *neural network* mengasilkan nilai 0.982 dengan nilai diagnosa *Excellent Classification*, sedangkan untuk algoritma *neural network* berbasis algoritma genetika menghasilkan nilai 0.971 dengan nilai diagnose *Excellent Classification*, dan selisih nilai keduanya sebesar 0.011.

4. Kesimpulan

Berdasarkan hasil eksperimen yang dilakukan dari hasil analisis optimasi model algoritma *neural network* berbasis algoritma genetika. Model yang dihasilkan diuji untuk mendapatkan nilai *accuracy*, *precision*, *recall* dan AUC dari setiap algoritma sehingga didapat pengujian dengan menggunakan *neural network* didapat nilai *accuracy* adalah 91.64 % dengan nilai *precision* 91.20 % dan nilai AUC adalah 0.942. sedangkan pengujian dengan menggunakan *neural network* berbasis *algoritma genetika* didapatkan nilai *accuracy* 93.03 % dengan nilai *precision* 91.28 % dan nilai AUC adalah 0.971. maka dapat disimpulkan pengujian model pemilu legislatif DKI Jakarta dengan menggunakan *neural network* dengan *neural network* berbasis *algoritma genetika* didapat bahwa pengujian *neural network* berbasis algoritma genetika lebih baik dari pada *neural network* sendiri.

Dengan demikian dari hasil pengujian model diatas dapat disimpulkan bawa *neural network* berbasis *algoritma genetika* memberikan pemecahan untuk permasalahan pemilu legislatif DKI Jakarta lebih akurat

Referensi

- Alejo RP, Trevino LT, Monorrez MP. 2008. Optimization Welding Process Parameters through Response Surface with Neural Network and Genetic Algorithm. Proceeding CERMA '08 Proceedings of the 2008 Electronics, Robotics and Automotive Mechanics Conference , 393-399 .
- Astuti E.D. 2009. Pengantar Jaringan Saraf Tiruan. Wonosobo: Star Publishing.
- Berndtssom M, Hansson J, Olsson B, Lundell B. 2008. A Guide for Students in Computer Science and Information Systems. London: Springer.
- Bidgoli BM, Punch WF. 2003. Using Genetic Algorithms for Data Mining Optimization in an Educational Web-based System. Proceeding GECCO'03 Proceedings of the 2003 international conference on Genetic and evolutionary computation: PartII , 2252-2263.
- Borisyuk R, Borisyuk G, Rallings C, Thrasher M. 2005. Forecasting the 2005 General Election:A Neural Network Approach. The British Journal of Politics & International Relations Volume 7, Issue 2 , 145–299.
- Choi JH, & Han ST. 1999. Prediction of Election Result using Descrimination of Non-Respondents: The Case of the 1997 Korea Presidential Election.
- Dawson CW. 2009. Projects in Computing and Information System A Student's Guide. England: Addison-Wesley.
- Gill GS. 2005. Election Result Forecasting Using two layer Perceptron Network. Journal of Theoretical and Applied Information Technology Volume 4 no.11 , 144-146.
- Gorunescu F. 2011. Data Mining Concept Model Technique. India: Springer.

- Gray DE. 2004. Doing Research in the Real World. New Delhi: SAGE.
- Han J, Kamber M. 2007. Data Mining Concepts and Technique. Morgan Kaufmann publisher.
- Haupt RL, Haupt SE. 2004. Practical Genetic Algorithm. Canada: A John Wiley & Sons.
- Ileana L, Rotar C, Incze A. 2004. The Optimization of Feed Forward Neural Networks Structure Using Genetic Algorithm. Proceeding IJCAI'89 Proceedings of the 11th international joint conference on Artificial intelligence - Volume 1 , 762-767 .
- K GS, Deepa DS. 2011. Analysis of Computing Algorithm using Momentum in Neural Networks. Journal of computing, volume 3, issue 6 , 163-166.
- Ke J, Liu X. 2008. Empirical Analysis of Optimal Hidden Neurons in Neural Network Modeling for Stock Prediction. 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application volume 02 , 828-832 .
- Kothari CR. 2004. Research Methodology methodes and Technique. India: New Age Interntional.
- Kusrini, Luthfi ET. 2009. Algoritma Data mining. Yogyakarta: Andi.
- Kusumadewi S, Purnomo H. 2005. Penyelesaian Masalah optimasi dengan teknik-teknik HEURISTIK. Yogyakarta: Graha Ilmu.
- Larose DT. 2006. Data Mining Methods and Models. Hoboken, New Jersey: John Wiley & Sons, Inc.
- Larose DT. 2005. Discovering Knowledge in Data. Canada: Wiley Interscience.
- Moscato P, Mathieson L, Mendes A, Berreta R. 2005. The Electronic Primaries: Prediction The U.S. Presidency Using Feature Selection with safe Data. ACSC '05 Proceedings of the Twenty-eighth Australasian conference on Computer Science - Volume 38 , 371-379 .
- Nagadevara, Vishnuprasad. 2005. Building predictive models for election results in India an application of classification trees and neural networks. Journal of Academy of Business and Economics volume 5 .
- Purnomo MH, Kurniawan A. 2006. Supervised Neural Network. Suarabaya: Garaha Ilmu.
- Rigdon SE, Jacobson SH, Sewell EC, Rigdon CJ. 2009. A Bayesian Prediction Model for the United State Presidential Election. American Politics Research July 2009 vol. 37 no. 4 , 700-724.
- Santoso T. 2004. Pelanggaran Pemilu dan penanganannya. Jakarta: The Habibie Center.
- Sardini NH. 2011. Restorasi Penyelenggaraan Pemilu di Indonesia. Yogyakarta: Fajar Media Press.
- Shukla A, Tiwari R, Kala R. 2010. Real Life Application of Soft Computing. CRC Press.
- Sug H. 2009. An Empirical Determination of Samples for Decision Trees. AIKED'09 Proceedings of the 8th WSEAS international conference on Artificial intelligence, knowledge engineering and data bases , 413-416 .

Undang-Undang RI No.10. (2008).

Vercellis C. 2009. Business Intelligence : Data Mining and Optimization for Decision Making. Southern Gate, Chichester, West Sussex: John Wiley & Sons, Ltd.

Xiao, Shao Q. 2011. Based on two Swarm Optimized algorithm of neural network to prediction the switch's traffic of coal. ISCCS '11 Proceedings of the 2011 International Symposium on Computer Science and Society , 299-302.