

Implementasi Algoritma Genetika pada Struktur *Backpropagation Neural Network* untuk Klasifikasi Kanker Payudara

Adam Mizza Zamani, Bilqis Amaliah dan Abdul Munif

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember

Jl. Arief Rahman Hakim, Surabaya 60111

E-mail: bilqis@its-sby.edu

Abstrak—Pada penelitian sebelumnya sudah dilakukan klasifikasi kanker payudara menggunakan kombinasi dari metode *Neural Network* (NN) dan *Association Rules* (AR). Namun metode ini dinilai masih belum optimal dikarenakan hasil rata-rata akurasi yang menunjukkan angka kurang maksimal. Metode baru diusulkan dengan maksud untuk mencari solusi lebih baik dari metode sebelumnya dalam hal rata-rata akurasi, yaitu dengan menggunakan kombinasi metode klasifikasi *Neural Network* dan algoritma genetika. Metode *Neural Network* digunakan sebagai *Artificial Intelligence* untuk memprediksi kanker payudara, sedangkan algoritma genetika digunakan untuk optimasi parameter *Neural Network* seperti jumlah *hidden layer* dan *learning rate* agar akurasi yang dihasilkan bisa lebih bagus. Selain itu, penelitian ini bertujuan untuk membandingkan antara metode kombinasi *Neural Network* dan algoritma genetika dengan metode *Naïve Bayesian*. Metode evaluasi uji coba yang digunakan adalah metode *10 fold cross validation*. Hasil uji coba *10 fold cross validation* menunjukkan bahwa metode *Neural Network* yang optimasi parameternya menggunakan algoritma genetika menghasilkan rata-rata akurasi yang cukup tinggi yaitu 97,00%, lebih baik dari metode *Naïve Bayesian* yang menghasilkan rata-rata akurasi 96,24% dan juga lebih baik dari metode *Neural Network* dengan *Association Rules* yang menghasilkan rata-rata akurasi 95.6%.

Kata Kunci—Algoritma genetika, *Backpropagation Neural Network*, *Cross Validation*, klasifikasi *Naïve Bayesian*.

I. PENDAHULUAN

Kanker payudara adalah jenis kanker paling umum yang diderita kaum wanita. Hingga saat ini, salah satu cara pengobatan yang umum dilakukan adalah dengan pembedahan dan jika perlu dilanjutkan dengan kemoterapi maupun radiasi. Namun pengobatan tersebut tidak akan memberikan dampak yang signifikan jika kankernya sudah mencapai stadium akhir. Oleh karena itu apabila penyakit ini dapat dideteksi lebih awal, dampak buruk yang mungkin ditimbulkan oleh kanker payudara dapat dicegah.

Untuk deteksi awal kanker payudara, pada penelitian sebelumnya sudah dilakukan klasifikasi kanker payudara menggunakan kombinasi dari metode *Neural Network* (NN) dan *Association Rules* (AR) [1]. Namun metode ini dinilai masih belum optimal dikarenakan hasil rata-rata akurasi yang menunjukkan angka kurang maksimal. Pada makalah ini akan diimplementasikan sebuah *Artificial Intelligence System* untuk membantu diagnosa awal kanker payudara yang menghasilkan rata-rata akurasi lebih baik dari metode sebelumnya. Metode tersebut adalah klasifikasi *Neural Network* menggunakan optimasi algoritma genetika [2].

Diketahui bersama bahwa prediksi yang dihasilkan dari model *Neural Network* akan lebih akurat jika parameter seperti jumlah unit lapisan tersembunyi (*hidden layer*) dan *learning rate* dapat dioptimalkan secara benar dan tepat. Untuk mencari efisiensi parameter tersebut diperlukan algoritma genetika. Prinsip dasar dari algoritma genetika sendiri adalah bagaimana mendapatkan keturunan yang lebih baik dengan melalui proses seleksi layaknya seleksi alam.

Selain membandingkan antara metode yang diusulkan dengan metode sebelumnya yaitu metode *Neural Network* ditambah dengan kombinasi *Association Rules* (AR), penulis juga coba membandingkannya dengan hasil akurasi yang diperoleh dari metode klasifikasi *Naïve Bayesian*. Dengan demikian, dari beberapa kali percobaan bisa disimpulkan metode manakah yang lebih baik dalam hal tingkat rata-rata akurasi.

II. OPTIMASI PARAMETER *NEURAL NETWORK* DENGAN MENGGUNAKAN ALGORITMA GENETIKA

Tujuan utama dari metode ini adalah harus mampu menemukan nilai akurasi terbaik model *Neural Network* dengan cara mengoptimasikan parameternya menggunakan algoritma genetika. Parameter pertama yang dioptimasi yaitu jumlah unit *hidden layer* yang dalam hal ini dibatasi satu *hidden layer*, dan parameter kedua yaitu *learning rate* (). Berikut ini akan dijelaskan secara ringkas langkah-langkah dari metode optimasi parameter *Neural Network* dengan algoritma genetika :

A. *Preprocessing dataset Breast Cancer*

Dari 699 *record*, terdapat 15 *missing value record*. *Missing value record* ialah *record* yang salah satu atau lebih atributnya tidak diketahui nilainya. Oleh karena itu perlu *preprocessing* data untuk menangani data yang *missing value* tersebut. *Preprocessing* data dilakukan dengan cara menghapus 15 *record* yang *missing value* tersebut sehingga dataset yang awalnya berjumlah 699 *record* tereduksi menjadi 684 *record*. Kelimabelas *record* itu masing-masing adalah 13 *record* dari kelas *Benign* (kelas 0), dan 2 *record* dari kelas *Malignant* (kelas 1). Pada akhirnya digunakan 684 *record* yang terdiri dari 445 *record* kelas *Benign* dan 239 *record* kelas *Malignant*.

B. *Pembangkitan populasi awal*

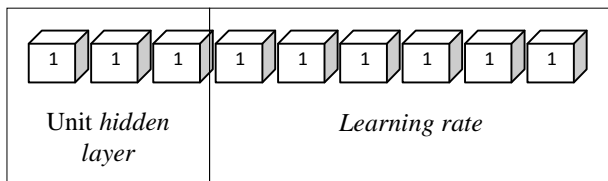
Tahap ini merupakan tahap pembentukan sejumlah kromosom dalam satu populasi. Kromosom direpresentasikan dalam bentuk bilangan biner 9 bit, masing-masing bit merupakan bilangan biner 0 dan 1. Tiga bit pertama

merepresentasikan jumlah unit pada *hidden layer*, sedangkan untuk enam bit setelahnya merepresentasikan nilai *learning rate* () [2]. Jumlah kromosom pada awalnya dibangkitkan secara acak yang ditentukan antara 10 sampai dengan 15 kromosom. Proses pembentukan satu buah kromosom dimulai dengan membangkitkan secara acak dua buah bilangan *k1* dan *k2*. Bilangan *k1* melambangkan jumlah unit pada *hidden layer*, sedangkan bilangan *k2* melambangkan *learning rate*. Kemudian kedua bilangan tersebut dikonversi ke dalam bentuk biner dengan panjang bit yang berbeda. Bilangan *k1* sebanyak 3 bit, sedangkan bilangan *k2* sebanyak 6 bit.

Hasil akhir berupa kromosom sebanyak 9 bit biner didapatkan dari penggabungan bilangan *k1* dan *k2* yang sudah dikonversi ke dalam bentuk biner.

C. Tahap pelatihan Neural Network

Pembentukan populasi yang dilakukan sebelumnya menghasilkan *n* kromosom yang tiap kromosomnya memiliki informasi parameter jumlah unit lapisan tersembunyi dan *learning rate*. Tujuan dari tahap pelatihan *Neural Network* ialah menghasilkan bobot *w* (bobot dari lapisan input ke lapisan tersembunyi) dan bobot *v* (bobot dari lapisan tersembunyi ke lapisan output) yang optimal dengan kedua parameter tersebut.



Gambar 1. Desain kromosom dengan 9 bit biner, merepresentasikan struktur dan parameter pelatihan *Neural Network*.

Selain kedua parameter tersebut, terdapat parameter lain yang digunakan yaitu lapisan input yang terdiri dari 9 unit, lapisan output yang terdiri dari 1 unit, bobot *w*, bobot *v*, dan kondisi berhenti *maxepoch* (maksimum iterasi pelatihan) yaitu sebanyak *n* kali. Algoritma yang digunakan pada saat pelatihan ialah algoritma *backpropagation* [3], [4], [5] dengan fungsi pada *hidden layer* dan *output layer* adalah *sigmoid biner*. Fungsi *sigmoid biner* didefinisikan menggunakan persamaan sebagai berikut :

$$y = \frac{1}{1 + e^{-u}} \tag{1}$$

u adalah nilai keluaran dari model, *y* berada pada *range* (0,1). Sebagai catatan fungsi ini hanya bisa dipakai pada kasus klasifikasi yang mana output target hanya dibedakan menjadi dua kelas target. Berikut akan dijelaskan secara ringkas algoritma *Backpropagation* :

- i. Inisialisasi bobot-bobot, konstanta laju pelatihan (), dan maksimal *epoch* (*n* kali pengulangan *learning* sebagai kondisi berhenti).
- ii. Selama kondisi berhenti belum dicapai, lakukan langkah ke-2 hingga langkah ke-9.
- iii. Untuk setiap pola pelatihan, lakukan langkah ke-3 sampai langkah ke-8.

Tahap feedforward

- iv. Setiap unit input *x_i* (dari unit ke-1 hingga unit ke-*n* pada lapisan input) mengirimkan sinyal input ke setiap input yang ada lapisan tersembunyi.
- v. Masing-masing unit lapisan tersembunyi *z_j* (dari unit ke-1 hingga unit ke-*p*) dikalikan dengan bobotnya (*w_{ji}*) dan dijumlahkan serta ditambahkan dengan bobot biasnya. Kemudian hasilnya dimasukkan ke dalam fungsi aktivasi.

$$z_net_j = w_{j0} + \sum_{i=1}^n x_i \cdot w_{ji} \tag{2}$$

$$z_j = f(z_net_j) = \frac{1}{1 + e^{-z_net_j}} \tag{3}$$

- vi. Masing-masing unit output *y_k* (*k* = 1,2,3,...*m*) dikalikan dengan bobot (*v_{ki}*) dan dijumlahkan serta ditambahkan dengan biasnya.

$$y_net_k = v_{k0} + \sum_{i=1}^p z_i \cdot v_{ki} \tag{4}$$

$$y_k = f(y_net_k) = \frac{1}{1 + e^{-y_net_k}} \tag{5}$$

Tahap backward

- vii. Masing-masing unit output *y_k* (*k* = 1,2,3,...*m*) menerima pola target *t_k* sesuai dengan pola input saat pelatihan dan kemudian informasi kesalahan atau error lapisan output (*δ_k*) dihitung. (*δ_k*) dikirim ke lapisan di bawahnya dan digunakan untuk menghitung besarnya koreksi bobot dan bias (*V_{jk}* dan *V_{ko}*) antara lapisan tersembunyi dengan lapisan output.

$$\delta_k = (t_k - y_k) y_k (1 - y_k) \tag{6}$$

Kemudian hitung suku perubahan bobot *V_{kj}* dan perubahan bias *V_{ko}* dengan laju pelatihan .

$$\Delta V_{kj} = \alpha \delta_k z_j ; k = 1, \dots, m ; j = 0, 1, \dots, p \tag{7}$$

$$\Delta V_{k0} = \alpha \delta_k \tag{8}$$

- viii. Pada setiap unit di lapisan tersembunyi (dari unit ke-1 hingga ke-*p*; *i*=1..*n*; *k*=1,..*m*) dilakukan perhitungan informasi kesalahan lapisan tersembunyi (*δ_j*). *δ_j* kemudian digunakan untuk menghitung besar koreksi bobot dan bias (*W_{ji}* dan *W_{jo}*) antara lapisan input dan lapisan tersembunyi.

$$\delta_net_j = \sum_{k=1}^m \delta_k v_{kj} \tag{9}$$

$$\delta_j = \delta_net_j z_j (1 - z_j) \tag{10}$$

Hitung suku perubahan bobot *W_{ji}* dan perubahan bias *W_{jo}* dengan laju pelatihan .

$$\Delta W_{ji} = \alpha \delta_j x_i \tag{11}$$

$$\Delta W_{j0} = \alpha \delta_j \tag{12}$$

Tahap perubahan bobot pada tiap lapisan dan bias

- ix. Masing-masing unit output *y_k* (*k*=1,2,3,...*m*) dilakukan perubahan bias dan bobotnya (*j*=0,1,2,..*p*) sehingga menghasilkan bobot dan bias yang baru :

$$v_{kj}(\text{baru}) = v_{kj}(\text{lama}) + \Delta v_{kj} \tag{13}$$

Demikian juga untuk setiap unit tersembunyi mulai dari unit ke-1 sampai unit ke-*p* dilakukan pengupdatean bobot dan bias.

$$w_{ji}(\text{baru}) = w_{ji}(\text{lama}) + \Delta w_{ji} \tag{14}$$

- x. Uji kondisi berhenti.

D. Tahap pengujian Neural Network

Bobot w dan bobot v dari tiap model kromosom *Neural Network* hasil tahap pelatihan sebelumnya digunakan sebagai parameter untuk menguji seberapa akurat prediksi yang dihasilkan model tersebut jika ada data uji baru yang masuk. Tahapan ini akan menghasilkan nilai y (output model) yang telah diaktivasi dengan fungsi *sigmoid biner*. Jika x adalah penjumlahan hasil perkalian *hidden layer* dengan masing-masing bobot yang berkoresponden dengan *output layer*-nya maka akan membentuk sebuah persamaan [6] :

$$y = f(x) \begin{cases} 1; x \geq 1 \\ 0.5 + x \cdot \left(1 - \frac{|x|}{2}\right); -1 \leq x \leq 1 \\ 0; x \leq -1 \end{cases} \quad (15)$$

Nilai y berkisar antara 0.0 sampai 1.0 untuk kemudian nilai nya dibandingkan dengan target pada data uji . Jika nilainya sama maka prediksi bernilai benar. Nilai akurasi didapatkan dari total prediksi benar model *Neural Network* dibagi dengan jumlah data uji lalu dikalikan dengan 100%. Akurasi itulah yang menjadi nilai *fitness* dari tiap kromosom.

E. Seleksi kromosom

Tahap seleksi bertujuan untuk mengetahui calon-calon kromosom mana yang memiliki nilai *fitness* optimal dalam satu populasi untuk kemudian dipilih dan dijadikan induk pada proses *crossover* (perkawinan silang). Metode seleksi yang digunakan adalah *Roulette Wheel Selection*. Dalam metode ini setiap kromosom atau individu memiliki nilai probabilitas yang jika digambarkan dalam diagram lingkaran setiap individu mendapatkan luas bagian sesuai dengan probabilitas nilai *fitness*-nya. Nilai probabilitas didapatkan dari *fitness* tiap kromosom dibagi total *fitness* dalam satu populasi.

Setelah nilai probabilitas masing-masing kromosom diketahui, maka langkah berikutnya adalah membentuk *chance table*. Tabel ini berisi nilai probabilitas batas bawah dan nilai probabilitas batas atas dengan panjang *range*-nya pun sesuai dengan nilai probabilitasnya. Kromosom mana yang akan terpilih ditentukan berdasarkan nilai r . Dimana nilai r adalah bilangan acak antara 0 s/d 1. Berada di posisi indeks manakah r dalam *chance table*, maka kromosom pada indeks itulah yang dipilih.

F. Perkawinan Silang (Crossover)

Tahap selanjutnya adalah proses *crossover*. *Crossover* dilakukan dengan melakukan pertukaran gen dari dua induk hasil tahap seleksi secara acak. Metode yang digunakan adalah *1-point crossover* yaitu memilih secara acak indeks diantara dua kromosom untuk dijadikan acuan *crossover*. Posisi indeks dipilih secara acak dengan banyak kemungkinannya adalah panjang bit kromosom dikurangi 1. Terjadi atau tidaknya proses ini dilakukan pada setiap kromosom, ditentukan oleh nilai probabilitas (cp) tertentu, sehingga dapat didefinisikan sebagai sebuah persamaan berikut [7] :

$$f(r) = \begin{cases} crossover, 0 \leq r \leq cp \\ not\ crossover, r \geq cp \end{cases} \quad (16)$$

r : bilangan acak antara 0 sampai 1.
 cp : *crossover probability*.

Jika kromosom sejumlah n semuanya mengalami *crossover*, maka jumlah *offspring* (anak kromosom baru) adalah $2n$. Semua *offspring* yang dihasilkan tersebut selanjutnya digunakan untuk proses mutasi gen.

G. Mutasi

Mutasi gen merupakan operator yang menukar nilai gen kromosom dengan nilai *invers*-nya. Misalnya nilai 0 menjadi 1 atau nilai 1 menjadi nilai 0. Terjadi atau tidaknya proses mutasi gen setiap kromosom ditentukan dengan probabilitas tertentu (mp). Untuk berapa bit yang dimutasi ditentukan secara acak dengan batasan panjang bit dari tiap kromosom. Posisi bit mana yang akan dimutasi juga ditentukan secara acak, sehingga dapat didefinisikan sebagai sebuah persamaan berikut [7] :

$$f(r) = \begin{cases} mutation, 0 \leq r \leq mp \\ not\ mutation, r \geq mp \end{cases} \quad (17)$$

r : bilangan acak antara 0 sampai 1.
 mp : *mutation probability*.

Kemudian dari hasil proses mutasi akan terbentuk populasi baru dimana nilai *fitness*-nya juga harus dihitung kembali menggunakan metode pelatihan dan pengujian *Neural Network* yang sudah dijelaskan sebelumnya. Dengan demikian diharapkan generasi baru yang muncul tersebut bisa lebih baik dari generasi sebelumnya dalam hal rata-rata akurasi tiap kromosom.

Selama kondisi berhenti belum terpenuhi, maka generasi baru akan terus dimunculkan. Berikut dua macam kondisi berhenti yang digunakan untuk membatasi generasi yang muncul, sehingga waktu komputasinya tidak terlalu lama :

- i. Kondisi berhenti (*stopping criteria*) yang pertama adalah maksimum jumlah generasi. Yaitu berhenti ketika mencapai generasi ke- n . Nilai n minimal 1, dan maksimal 100.
- ii. Kondisi berhenti (*stopping criteria*) yang kedua adalah konvergensi. Yaitu berhenti ketika nilai rata-rata *fitness* pada 5 generasi terakhir tidak berubah atau konvergen.

H. Naïve Bayesian

Naïve Bayesian merupakan salah satu metode klasifikasi berdasarkan penerapan teorema *bayes* dan masuk ke dalam jenis *supervised learning*. Untuk estimasi parameternya menggunakan *maximum likelihood* dengan hanya dua parameter yang digunakan yaitu *mean* dan *variance* [8]. Langkah pertama ialah mencari probabilitas tiap atribut terhadap masing-masing kelas dengan menggunakan persamaan distribusi normal. Misalkan diasumsikan atribut dari sebuah data pelatihan adalah x , μ_c adalah *mean* dari tiap-tiap atribut yang berasosiasi dengan kelas c , sedangkan σ_c^2 adalah *variance*-nya, maka nilai probabilitas $P(x = v|c)$ dapat didefinisikan sebagai sebuah persamaan :

$$P(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}} \quad (18)$$

Kemudian mencari *posterior* dari masing-masing kelas, dimana kelas dengan *posterior* lebih tinggi mengindikasikan bahwa data tersebut masuk ke dalamnya. Untuk mencari *posterior* dapat didefinisikan sebagai sebuah persamaan :

$$\text{post}(c|x_1, x_2, \dots) = P(x_1|c) \cdot P(x_2|c) \cdot P(x_n|c) \cdot P(c) \quad (19)$$

III. UJI COBA

A. Data Uji Coba

Data yang digunakan dalam uji coba ini adalah dataset *Breast Cancer Wisconsin* [9]. Dataset tersebut dibagi dengan menggunakan metode *10-fold cross validation* [8]. *10-fold cross validation* adalah salah satu metode validasi uji coba. Dataset yang awalnya berjumlah 684 data akan dibagi menjadi 10 *fold* (bagian). Kombinasi 9 *fold* yang berbeda akan digabung dan digunakan sebagai data pelatihan, sedangkan 1 *fold* (sisa) digunakan sebagai data uji.

Masing-masing *fold* berjumlah 68 data (684 dibagi 10 sisa 4 data). 4 data yang tersisa digunakan untuk tambahan data uji pada setiap *fold*-nya, artinya 4 sisa data tersebut selalu ditambahkan pada *fold* yang berfungsi sebagai data uji.

B. Skenario Uji Coba

Uji coba dibagi menjadi 4 skenario utama. Masing-masing skenario memiliki tujuan yang berbeda dalam mengukur kinerja metode. 4 skenario tersebut adalah :

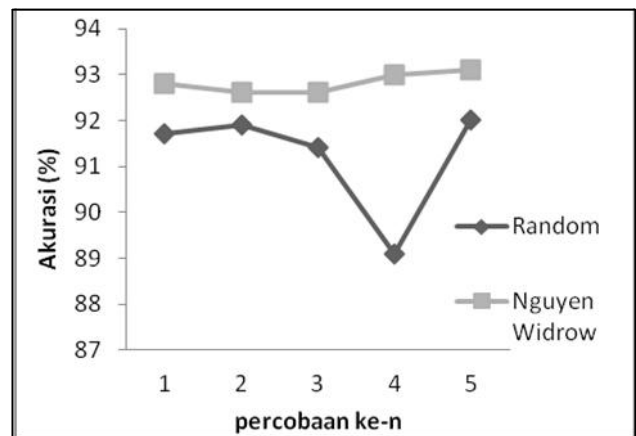
- 1) Uji performa dengan pembandingan *stopping criteria* pada algoritma genetika. Tujuan utama dari pengujian ini adalah membandingkan *stopping criteria* mana yang menghasilkan nilai rata-rata maksimum akurasi terbaik. Dalam hal ini, *stopping criteria* yang dibandingkan adalah *100 maximum generation* dengan *konvergent generation*. Pada skenario 1 ini nilai *crossover probability* yang dipakai adalah 0,0-0,75, sedangkan inialisasi bobot awal *Neural Network* dibangkitkan secara acak.
- 2) Uji performa dengan pembandingan inialisasi awal bobot pada model *Neural Network*. Tujuan utama dari pengujian ini adalah membandingkan metode inialisasi bobot awal pada *Neural Network* yang menghasilkan nilai rata-rata maksimum akurasi terbaik. Dalam hal ini, metode yang dibandingkan adalah inialisasi bobot secara *random* dengan metode *Nguyen-Widrow*. Pada skenario 2 ini nilai *crossover probability* yang dipakai adalah 0,0-0,75, sedangkan *stopping criteria*-nya sebanyak 100 generasi.
- 3) Uji performa dengan pembandingan *cp* (*crossover probability*) pada algoritma genetika. Tujuan utama dari pengujian ini adalah mencari *range* nilai *cp* (*crossover probability*) yang menghasilkan nilai rata-rata maksimum akurasi terbaik. Dalam hal ini, *range* nilai *cp* yang diuji pertama yaitu : 0-0,5, kedua yaitu : 0-0,75, dan ketiga yaitu : 0-0,1. Pada skenario 3 ini *stopping criteria* nya sebanyak 100 generasi. Sedangkan untuk inialisasi bobot awal *Neural Network* menggunakan metode *Nguyen Widrow*.
- 4) Uji stabilitas performa dengan metode *10 fold cross validation*. Skenario ini bertujuan untuk menguji stabilitas nilai akurasi jika diuji dengan data yang berbeda. Dikatakan berhasil jika nilai rata-rata standar deviasi dari tiap pengujian *fold* data bernilai kecil. Untuk skenario 4

ini digunakan nilai *cp* (*crossover probability*) 0-0,75, dan inialisasi bobot awal *Neural Network* menggunakan metode *Nguyen Widrow*. Sedangkan untuk *stopping criteria*-nya menggunakan kriteria maksimum 100 generasi. Pemilihan *stopping criteria* tersebut dimaksudkan agar ketidakseimbangan nilai akurasi bisa dihindari, sebab jumlah generasi mempengaruhi terhadap rata-rata akurasi pada tiap pengujian data.

C. Hasil dan Analisis Uji Coba

Pada skenario uji coba pertama didapatkan hasil rata-rata sebesar 98,4% dan standar deviasi sebesar 0,013 untuk kriteria *maximum generation*. Sedangkan untuk kriteria konvergensi didapatkan hasil rata-rata sebesar 99,0% dan standar deviasi sebesar 0,010. Hal ini menunjukkan hasil rata-rata konvergensi lebih baik dari pada hasil rata-rata generasi maksimum walaupun perbedaannya terpaut sangat sedikit. Nilai standar deviasi dari kedua kriteria menunjukkan bahwa nilai akurasi tiap generasi memiliki homogenitas tinggi, artinya persebaran data tidak terlalu jauh dan terpusat di atas 90% (*mean*).

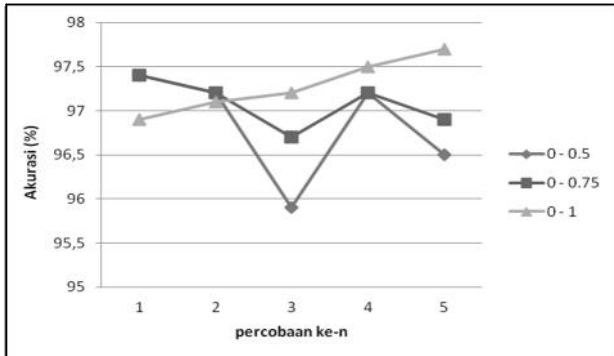
Pada skenario uji coba kedua mengindikasikan nilai rata-rata untuk inialisasi bobot awal *neural network* dengan metode *Nguyen Widrow* lebih baik dibandingkan inialisasi bobot awal *neural network* secara acak. Nilai rata-rata standar deviasi untuk masing-masing metode hanya terpaut sekitar 0,005. Untuk inialisasi bobot awal secara acak, didapatkan rata-rata akurasi 92.5% dan standar deviasi sebesar 0.018. Sedangkan untuk inialisasi bobot awal dengan metode *Nguyen Widrow*, didapatkan rata-rata akurasi **93.4%** dan standar deviasi sebesar **0.013**. Hal ini menunjukkan persebaran data kedua metode memiliki tingkat homogenitas relatif tinggi, yang berarti stabilitas nilai akurasi terjaga. Hasil uji coba untuk skenario kedua dapat dilihat pada Gambar 2.



Gambar 2. Hasil 5 kali percobaan perbandingan inialisasi bobot awal *Neural Network* dengan metode *Nguyen Widrow* dan inialisasi bobot awal *Neural Network* secara acak.

Pada skenario uji coba ketiga mengindikasikan nilai rata-rata akurasi semakin meningkat berbanding lurus dengan kenaikan nilai *cp*. Untuk *cp* 0-0.5 menghasilkan rata-rata akurasi sebesar 96.84% dan standar deviasi sebesar 0,018, lalu untuk *cp* 0-0.75 menghasilkan rata-rata akurasi sebesar 97.08% dan standar deviasi sebesar 0,009. Sedangkan untuk

cp 0-1.0 menghasilkan rata-rata akurasi sebesar 97.28% dan standar deviasi sebesar 0,010. Dapat ditarik kesimpulan bahwa semakin tinggi nilai *cp* nya, maka kemungkinan rata-rata akurasi juga semakin meningkat. Gambar 3 adalah hasil uji coba untuk skenario ketiga yang disajikan dalam bentuk grafik titik.



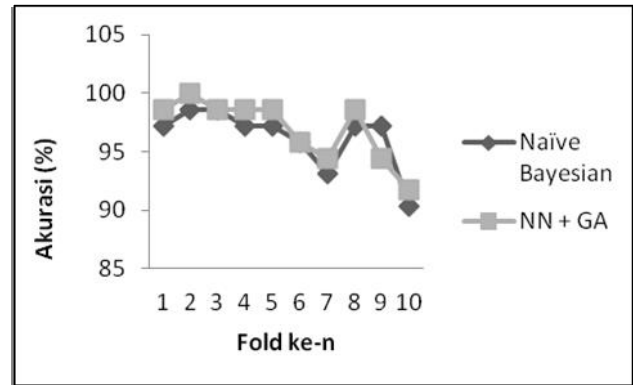
Gambar 3. Hasil 5 kali percobaan perbandingan nilai *cp* (crossover probability).

Tabel 1. Hasil uji coba skenario ke-4 yaitu 10 fold cross validation

Fold	Rata-rata akurasi (%)	Std.dev
1	98,6	0,01
2	100	0,01
3	98,6	0,01
4	98,6	0,08
5	98,6	0,02
6	95,8	0,01
7	94,4	0,01
8	98,6	0,01
9	94,4	0,01
10	91,7	0,03
Rata-rata	97,0	0,03

Hasil pada Tabel 1 mengindikasikan bahwa nilai akurasi pada tiap *fold* sudah sangat stabil. Percobaan pada *fold* ke-10 memperlihatkan hasil paling rendah diantara 9 *fold* lainnya yaitu 91,7%, sedangkan percobaan pada *fold* ke-2 memperlihatkan hasil paling tinggi diantara 9 *fold* lainnya yaitu 100%. Jika dianalisis, performa model *Neural Network* yang dihasilkan sudah bisa dikatakan stabil. Pernyataan tersebut diperkuat dengan nilai standar deviasi dari tiap *fold* yang sangat kecil yaitu 0.04. Nilai itu mengindikasikan kalau persebaran atau rentang nilai akurasi cenderung bersifat homogen.

Hasil skenario uji coba keempat yang ditunjukkan pada Gambar 4 menunjukkan rata-rata akurasi metode kombinasi *Neural Network* dengan optimasi parameter menggunakan algoritma genetika lebih baik dari rata-rata akurasi metode *Naive Bayesian* walaupun perbedaannya sangat tipis, yaitu 0,008.



Gambar 4. Hasil perbandingan uji coba 10 fold cross validation antara metode *Neural Network* yang dioptimasi menggunakan algoritma genetika dengan metode *Naive Bayesian*.

IV. KESIMPULAN

Metode *Neural Network* yang parameternya dioptimalkan menggunakan algoritma genetika terbukti mampu menghasilkan nilai rata-rata akurasi yang tinggi yaitu sebesar 97,00% untuk studi kasus deteksi kanker payudara. Metode ini memiliki keunggulan dalam hal tingkat akurasi jika dibandingkan metode *Naive Bayesian* dan metode sebelumnya yaitu *Neural Network* menggunakan optimasi *Association Rules*. Namun metode ini masih mempunyai beberapa kelemahan diantaranya adalah membutuhkan waktu komputasi yang lama. Untuk meningkatkan performa akurasi, dapat dipilih parameter optimal untuk *cp* (crossover probability) antara 0.0-1.0, sedangkan penggunaan metode *Nguyen Widrow* untuk inialisasi bobot awal *Neural Network* terbukti menghasilkan rata-rata akurasi lebih baik dari pada inialisasi bobot awal *Neural Network* secara acak.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Allah SWT atas limpahan rahmat dan hidayah-Nya dan semua pihak yang telah membantu dalam penulisan makalah ini.

DAFTAR PUSTAKA

- [1] Murat Karabatak , M. Cevdet Ince, An expert system for detection of breast cancer based on association rules. *Expert Systems with Applications*, (2009).
- [2] Paul S. Heckerlinga, Ben S. Gerbera, Thomas G. Tapece, Robert S. Wigton, Use of genetic algorithms for neural networks to predict community-acquired pneumonia. *Artificial Intelligence in Medicine*, (2004).
- [3] Chakraborty, R. (01 Juni 2010). *Backpropagation Network : Soft Computing Course Lecture 15-20*. Diakses tanggal 30 Maret 2012, dari http://www.myreaders.info/03_Back_Propagation_Network.pdf
- [4] Chakraborty, R. (01 Juni 2010). *Fundamentals of Neural Network : AI Course lecture 37-38*. Diakses tanggal 30 Maret 2012, dari http://www.myreaders.info/08_Neural_Networks.pdf
- [5] Ivan Siregar. *Jaringan Syaraf Tiruan*. Diakses tanggal 27 Maret 2012, dari http://ivan.siregar.biz/courseware/CG2NeuralNetwork_Algorithm.pdf
- [6] Tveter, D. R. (18 Nopember 2001). Diakses tanggal 01 April 2012, dari Backpropagator's Review: <http://www.dontveter.com/bpr/activate.html>

- [7] Chakraborty, R. (01 Juni 2010). *Fundamentals of Genetic Algorithms : AI Course Lecture 39-40*. Diakses tanggal 30 Maret 2012, dari http://www.myreaders.info/09_Genetic_Algorithms.pdf
- [8] Payam Refaeilzadeh, Lei Tang, Huan Liu, *Cross-Validation*. Arizona State University, (2008).
- [9] Dr. William H. Wolberg. (08 Januari 1991). *Wisconsin Breast Cancer Database*. Diakses tanggal 20 Maret 2012, dari http://archive.ics.uci.edu/ml/machine-learning_databases/breast-cancer-wisconsin/breast-cancer_wisconsin.data.