PEMBELAJARAN BERTINGKAT PADA ARSITEKTUR JARINGAN SARAF FUNGSI RADIAL BASIS

Diana Purwitasari[1], Glory Intani Pusposari[2], Rully Sulaiman[3]
Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya
Jl. Raya ITS - Gedung Teknik Informatika ITS Surabaya 60111, Telp : +62 (031)5939214
E-mail : diana@if.its.ac.id

ABSTRAK

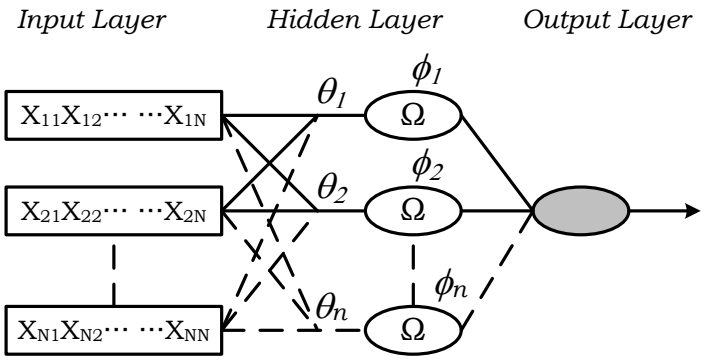
Jaringan saraf tiruan (JST) adalah jaringan yang cara kerjanya meniru jaringan saraf manusia ditandai dengan sebuah set masukan dan sebuah set keluaran. Proses pembelajaran dalam jaringan akan mengekstraksi informasi dari berbagai macam input yang diberikan. Diantara masukan dan keluaran terdapat layer untuk memproses input yang dinamakan unit tersembunyi (hidden layer). Salah satu model JST adalah jaringan saraf fungsi radial basis (Radial Basis Function Neural Network = RBFNN) yaitu model jaringan saraf dengan satu unit dalam lapisan tersembunyi. Jumlah layer tunggal pada hidden layer menyebabkan permasalahan pembelajaran di RBFNN dapat dianggap sebagai suatu sistem linear. Pada RBFNN fungsi aktivasi yang digunakan adalah fungsi basis (Gaussian) dengan fungsi linear di lapisan output. Dikarenakan RBFNN adalah sistem linear sehingga teknik Orthogonal Least Squares (OLS) yang menerapkan konsep basis orthogonal dengan pendekatan terdekat ke solusi sebenarnya dapat menjadi salah satu algoritma pembelajaran pada RBFNN. Makalah ini membahas pembelajaran bertingkat sebagai cara optimasi pembelajaran pada RBFNN yang menggabungkan teknik linear yaitu Regularized Orthogonal Least Sqaures (ROLS) dan non linear yaitu algoritma genetik. Hasil ujicoba menunjukkan untuk semua data dengan persentase pembelajaran dan parameter algoritma genetik yang berbeda-beda mempunyai akurasi yang bervariasi pula. Akan tetapi rata-rata hasil ujicoba menghasilkan akurasi diatas 90% dan bahkan untuk beberapa percobaan akurasi bisa mencapai 100%.

Kata kunci : jaringan saraf fungsi radial basis, optimasi pembelajaran, regularized orthogonal least sqaures, algoritma genetik

1. PENDAHULUAN

Jaringan saraf fungsi radial basis (*Radial Basis Function Neural Network*, RBFNN) adalah suatu jenis arsitektur jaringan saraf tiruan, yakni jaringan dengan cara kerja meniru jaringan saraf manusia dan terdiri dari berlapis-lapis neuron yang bekerja bersama-sama untuk memecahkan suatu permasalahan. Jaringan saraf fungsi radial basis juga memiliki topologi jaringan seperti jaringan saraf tiruan yang lain terdiri atas unit masukan (*input layer*), unit tersembunyi (*hidden layer*), dan unit keluaran (*output layer*) [1]. Jaringan saraf fungsi radial basis adalah jaringan saraf *feed-forward* bersifat khusus yakni: (a) proses antara *input layer* ke *hidden layer* adalah nonlinier sedangkan proses antara *hidden layer* ke *ouput layer* bersifat linear; (b) fungsi aktivasi pada *hidden layer* berbasis radial seperti fungsi Gaussian; dan (c) *output layer* merupakan hasil penjumlahan.

RBFNN dapat diaplikasikan ke berbagai domain permasalahan antara lain seperti pemodelan data time-series, pengklasifikasian, pengenalan suara, restorasi gambar, estimasi gerak dan segmentasi benda bergerak. Makalah ini membahas penggunaan RBFNN dalam problem pengklasifikasian [2]. Node-node pada *input layer* merepresentasikan fitur-fitur dari data sedangkan node pada *output layer* memiliki keterkaitan dengan kelas pada pengklasifikasian. Implementasi model pengklasifikasian RBFNN yang memiliki satu *input layer*, satu *output layer* serta diantaranya terdapat *hidden layer* dengan pengaktifannya menggunakan fungsi Gaussian ditunjukkan pada Gambar 1. Seperti arsitektur jaringan saraf pada umumnya, RBFNN dapat digunakan untuk pembelajaran pola klasifikasi yaitu proses mengelompokkan data pada kelas-kelas tertentu yang telah diketahui sebelumnya.



Gambar 1: Arsitektur RBFNN untuk pengklasifikasian

Orthogonal Least Squares (OLS) merupakan salah satu prosedur yang sering digunakan dalam pembelajaran RBFNN dengan menerapkan konsep basis orthogonal dalam menyelesaikan sistem linear. OLS melakukan pendekatan terdekat ke solusi sebenarnya sehingga dapat menemukan bobot jaringan dengan nilai kesalahan sekecil mungkin. Jika data untuk proses pembelajaran memiliki *noise* terlalu tinggi, maka kombinasi regularisasi dengan OLS, *Regularized Orthogonal Least Squares* (ROLS) dapat menjamin adanya generalisasi pada pembelajaran. Penggunaan metode ROLS bertujuan untuk menemukan bobot jaringan saraf fungsi radial basis dengan nilai *error* terkecil menggunakan prinsip *least squares* untuk meminimalkan nilai *error* tersebut.

Untuk meningkatkan hasil pembelajaran, ROLS akan dikombinasikan dengan Algoritma Genetik. Algoritma tersebut dapat memberikan nilai bobot mendekati optimal pada proses pembelajaran, meskipun kelemahannya adalah permintaan akan biaya komputasi yang cukup besar. Penentuan bobot jaringan dengan ROLS akan dibatasi oleh suatu kriteria regularisasi dengan penggunaan algoritma genetik pada beberapa parameter mengingat kelemahan algoritma tersebut. Tujuan yang ingin dicapai disini adalah cara menerapkan sebuah metode pembelajaran bertingkat untuk RBFNN yaitu dengan kombinasi ROLS dan optimisasi algoritma genetik [2, 3, 4].

2. RBFNN PADA MODEL KLASIFIKASI

Standar arsitektur jaringan RBFNN memiliki tiga lapisan: *input*, *hidden*, dan *output*. RBFNN yang digunakan mempunyai satu *output* dan sebuah *nonlinear* Gaussian dengan varians data atau lebar (*width*) dibuat seragam, dan kemudian disebut dengan ρ. RBFNN memakai fungsi exponensial dari Gaussian guna membangun pendekatan lokal pada pemetaan *nonlinear input* dan *output*. Selanjutnya *output* yang ditunjukkan pada Gambar 1 didefinisikan pada persamaan (1)

$$\hat{y} = \sum_{i=1}^{n} \theta_i \, exp\left(\frac{-\|x-c_i\|^2}{\rho}\right) \qquad (1)$$

dengan *x* adalah *input vector* dari jaringan, *n* adalah jumlah *hidden unit*, θ_i adalah bobot *hidden unit* ke-*i*, c_i adalah *vector center* ke-*i*, dan $\|.\|$ menandakan penggunakan jarak *Euclidean*. Jumlah *hidden unit* menandakan jumlah kelas yang ada dalam pengaplikasian RBFNN untuk kasus klasifikasi.
Proses pembelajaran RBFNN ini akan diterapkan pada klasifikasi data sejumlah *N* terdiri dari matriks $y_{(k)}$ sebagai *output* dan $x_{(k)}$ sebagai *input*, dengan *k = 1, 2, ..., N*. Apabila data $x_{(k)}$diklasifikasikan dalam kelas *i* maka nilai $y_{(i)} = 1$ dan $y_{(\neq i)} = 0$. Berdasarkan data tersedia dapat ditentukan nilai *center* c_i yang merupakan rata-rata dari tiap fitur pada masing-masing kelas klasifikasi.
Fungsi aktivasi Gaussian pada RBFNN terkait dengan data *input* dinyatakan dalam persamaan (2).

$$\phi_{i_{(k)}} = exp\left(\frac{-\|x_{(k)}-c_i\|^2}{\rho}\right) \quad (2)$$

sehingga output RBFNN $y_{(k)}$ menjadi dinyatakan dengan persamaan (3).

$$y_{(k)} = \hat{y}_{(k)} + e_{(k)} = \sum_{i=1}^{n} \theta_i \, \phi_{i_{(k)}} + e_{(k)} \quad (3)$$

dengan $e_{(k)}$ adalah *error* diantara *output* jaringan yang diharapkan $y_{(k)}$ dan *output* jaringan yang sesungguhnya $\hat{y}_{(k)}$. Komputasi matriks persamaan (3) akan menjadi persamaan (4) yang merupakan model regresi linear dengan *least square*.

$$y = \Theta\Phi + e \quad (4)$$

Berdasarkan definisi dari

$$y = [y_{(1)}, y_{(2)}, \dots, y_{(N)}]^T \quad (5)$$

$$e = [e_{(1)}, e_{(2)}, \dots, e_{(N)}]^T \quad (6)$$

$$\Theta = [\theta_{(1)}, \theta_{(2)}, \dots, \theta_{(N)}]^T \quad (7)$$

$$\phi_i = [\phi_{i_{(1)}}, \phi_{i_{(2)}}, \dots, \phi_{i_{(N)}}]^T \quad (8)$$

$$\Phi = [\phi_{(1)}, \phi_{(2)}, \dots, \phi_{(N)}] \quad (9)$$

Nilai *y* langsung dapat dihitung dari data. Kemudian untuk menghitung matriks Φ berdasarkan persamaan (2), terlebih dahulu perhitungan jarak Euclidean harus dilakukan dengan memastikan *x* dan c_i mempunyai jumlah baris yang sama. Langkah selanjutnya adalah menemukan bobot Θ pada arsitektur RBFNN dengan *error e* sekecil mungkin (lihat persamaan (4)).

Output RBFNN (persamaan (4)) adalah linear, sehingga pencarian solusi akan menggunakan sistem linear. Untuk menekan tingkat kesalahan pada sistem dalam mencari solusi sistem persamaan yang paling mendekati solusi sebenarnya, model matriks dari persoalan akan didekomposisi pada basis matrik yang paling optimal yaitu matriks yang semua vektornya *orthogonal* terhadap yang lain. Sebagai akibatnya *error* pembelajaran RBFNN juga akan ikut ditekan.

Ada tiga persoalan yang harus dapat dipecahkan untuk mencari solusi persamaan (4) dengan model matriks: (i) masalah matriks singular yakni determinan matriks = 0, (ii) *undetermined system* yakni jumlah persamaan < variable yang tidak diketahui, dan (iii) *overdetermined system* yakni jumlah persamaan > variabel yang tidak diketahui.

Untuk mencari nilai matriks Θ yang dapat mengatasi kesemua persoalan tersebut akan digunakan *pseudoinvers*. Metode tersebut mencari solusi dengan *error e* yang paling minimum, yaitu jarak Euclidean antara *y* dan $\Theta\Phi$ memiliki jarak paling minimal. *Least Squares* adalah metode yang sering digunakan untuk meminimalkan *error*. Dekomposisi QR dapat dilakukan dengan terlebih dahulu melakukan proses Gram Schmidt untuk memastikan matriks dengan basis sembarang menjadi berbasis *orthonormal*. Sehingga sistem permasalahan pada persamaan (4) menjadi persamaan (10) dengan matriks *W* berupa nilai *eigenvector* serta *A* berupa *eigenvalue*.

$$y = \Theta\Phi + e = Wg + e = WA\Theta + e \quad (10)$$

Jika data pembelajaran memiliki *noise* terlalu banyak, maka pendekatan *least square* OLS tidak mampu menjamin generalisasi pada pengujian. Pendekatan regularisasi pada OLS, disebut ROLS, digunakan untuk meningkatkan generalisasi agar hasil pengujian stabil dan tidak terpengaruh dengan keberadaan *noise*. Selanjutnya penentuan bobot jaringan akan dibatasi kriteria regularisasi J_R yang dinyatakan pada persamaan:

$$J_R(g;\lambda) = e^T e + \lambda g^T g \quad (11)$$

Notasi λ adalah parameter regularisasi dengan criteria $\lambda \geq 0$ sedemikian hingga bobot yang ditemukan harus mampu meminimalkan J_R. Algoritma genetik digunakan untuk optimasi bobot yang meminimalkan J_R.

3. ALGORITMA PEMBELAJARAN BERTINGKAT

Dua level pembelajaran jaringan fungsi basis radial yang diimplementasikan adalah penggunaan algoritma genetik pada level atas dan metode ROLS pada level bawah. Kegunaan utama dari penggunaan algoritma genetik sebagai metode pembelajaran adalah kemampuannya untuk mencapai topologi jaringan dan pemberian nilai bobot yang optimal atau mendekati optimal. Dikarenakan performansi generalisasi yang dibatasi oleh kriteria regularisasi *error* J_R adalah fungsi yang kompleks pada nilai *varians* ρ dan parameter regularisasi λ maka dua nilai tersebut akan dioptimasi dengan algoritma genetik. Sedangkan tujuan penggunaan pembelajaran dengan ROLS adalah menemukan bobot θ_i yang paling meminimalisasi *error*.

Praproses data terjadi sebelum melakukan langkah-langkah pembelajaran bertingkat pada arsitektur RBFNN. Praproses meliputi langkah untuk memisahkan data menjadi data pembelajaran dan data pengujian. Kemudian mengenali kelas yang tersedia dalam suatu data standar, mengurutkan input berdasarkan kelas, serta mencari *center* c_i pada setiap kelas. Selanjutnya untuk tiap kelas akan dijalankan algoritma genetik yang memakai nilai dari *center* kelas tersebut.

Langkah-langkah pembelajaran dengan algoritma genetik adalah sebagai berikut [5]:

1. Inisialisasi populasi awal dengan kromosom sepanjang 16 bit (8 bit pertama untuk representasi nilai *varians* ρ pada fungsi aktivasi Gaussian, dan 8 bit selanjutnya untuk parameter regulasi λ). Keduanya memiliki batasan nilai; $0 \leq \rho \leq 1$ dan $0 \leq \lambda \leq 1$.
2. Proses evaluasi untuk menghitung nilai *fitness* dengan pendekatan ROLS membutuhkan komputasi dengan persamaan (4) yang bergantung pada hasil komputasi persamaan (2) dan (3). Nilai-nilai yang didapat akan digunakan dalam perhitungan kriteria regularisasi *error* J_R (persamaan (11)) sebagai nilai *fitness*. Sehingga proses evaluasi tidak hanya akan menghasilkan nilai evaluasi (*fitness*), namun juga mendapatkan nilai bobot jaringan dan nilai *error*. Sebagai catatan, pada umumnya algoritma genetik digunakan untuk fungsi memaksimalkan sesuatu. Namun optimasi parameter regularisasi λ bertujuan untuk meminimalkan sehingga nilai asli dari nilai *fitness* akan dikali dengan -1 terlebih dahulu.
3. Prosedur seleksi, mutasi, dan operator genetik *crossover* dilakukan untuk menentukan populasi pada iterasi berikut.

Langkah-langkah algoritma pembelajaran bertingkat dengan arsitektur jaringan RBFNN pada pengklasifikasian adalah: (i) Menentukan nilai untuk bermacam-macam parameter yang digunakan algoritma genetik (jumlah populasi, nilai probabilitas operator genetik seperti *crossover* dan mutasi, jumlah *crosspoint*, *threshold*, jumlah kemunculan nilai *threshold* sama, jumlah iterasi). Pembelajaran dengan algoritma genetik akan berhenti jika nilai fungsi *fitness* mencapai konvergen yaitu perubahan nilai *fitness* tidak melebihi *threshold* sebanyak jumlah sama secara berturut-turut atau iterasinya telah mencapai jumlah

iterasi tertentu. (ii) Menentukan nilai *center* pada setiap kelas. (iii) Melakukan pembelajaran dengan algoritma genetik untuk mendapatkan nilai *varians* dan parameter regularisasi yang optimal. (iv) Menghitung model linear dari RBFNN dengan pembelajaran ROLS untuk menentukan nilai bobot. (v) Melakukan pengujian akurasi kebenaran berdasarkan model linear yang sudah ditentukan parameter – parameternya melalui pembelajaran bertingkat.

4. UJI COBA

Uji coba dilakukan guna melihat akurasi hasil pengklasifikasian RBFNN dengan nilai parameter yang di diestimasi melalui pembelajaran bertingkat dari ROLS dan algoritma genetik. Data yang digunakan adalah data standar untuk kasus pengklasifikasian dari UCI Machine Learning Repository (data IRIS, data LENSA, data WINE, dan data GLASS) pada url http://archive.ics.uci.edu/ml/ [6]. Data IRIS adalah data 150 bunga dengan 4 fitur (panjang dasar bunga atau *sepal length*, lebar dasar bunga atau *sepal width*, panjang daun bunga atau *petal length* dan lebar daun bunga atau *petal width*) dan 3 kelas terbagi rata (*Iris Setosa* – 50 data, *Iris Virginica* – 50 data dan *Iris Versicolor* – 50 data). Data LENSA berisi 24 data kontak lensa dengan 4 fitur dan 3 kelas (*hard contact lenses* – 4 data, *soft contact lenses* – 5 data dan bukan keduanya – 15 data). Data WINE terdiri dari 178 data jenis anggur dengan 13 fitur dan 3 (59 – 71 – 48 data) kelas yang menggunakan analisa kimia sebagai fitur untuk menentukan asal anggur. Data GLASS terdiri dari 214 data dengan 9 fitur dan 7 (70 – 76 – 17 – 0 – 13 – 9 – 29 data) kelas.

Input yang dibutuhkan pada uji coba dibedakan menjadi dua kelompok yaitu *input* data umum dan *input* untuk parameter algoritma genetik. Untuk pengambilan data pelatihan, sebagai contoh *input* data umum, sebelumnya semua data akan diurutkan, kemudian data tiap kelas akan diambil sejumlah porsentase data pelatihan serta sisanya akan menjadi data pengujian.

Untuk parameter algoritma genetik perlu dilakukan penentuan nilai – nilai parameter yang ditunjukkan pada Gambar 2: (1) jumlah populasi kromosom pada inisialisasi, nilai probabilitas maksimum terjadinya (2) *crossover* dan (3) mutasi gen, (4) jumlah iterasi maksimum dalam *running* algoritma genetik, (5) jumlah titik untuk pemotongan kromosom saat dilakukan *crossover*, (6) nilai *threshold* sebagai ambang batas perubahan nilai *fitness* antar iterasi, serta (7) jumlah iterasi dengan nilai *fitness* yang sama sebagai tanda bahwa kondisi konvergen telah tercapai. Skenario untuk uji coba dengan berbagai variasi nilai parameter algoritma genetik ditunjukkan pada Tabel 1.

Uji coba untuk setiap skenario dilaksanakan sebanyak lima kali yang kemudian dilakukan penghitungan rata-rata, standar deviasi dan selang kepercayaan guna menentukan akurasi per skenario. Iterasi pada algoritma genetik akan berulang atau kondisi konvergen berarti belum tercapai, jika selisih nilai *fitness* dari satu iterasi ke iterasi berikut lebih kecil dari nilai *threshold* secara berturut-turut selama sejumlah *n* kali iterasi sesuai dengan ketentuan pada skenario. Apabila kondisi tersebut tidak terpenuhi, proses akan dihentikan saat jumlah iterasi telah mencapai 100. Pada satu proses pembelajaran algoritma genetik, perbandingan nilai *fitness* antar iterasi juga dilakukan untuk penentuan populasi pada iterasi selanjutnya. Jika nilai *fitness* saat inisialisasi lebih besar, maka populasi awal akan digunakan untuk iterasi berikutnya.

Gambar 2: Antarmuka untuk memasukkan inisialisasi nilai parameter pada pembelajaran bertingkat

Tabel 1: Variasi nilai parameter pada uji coba

Skenario	*% Data pembelajaran*	*Populasi*	*% Crossover*	*Crosspoint*	*Threshold*	*Jml. iterasi utk. Konvergen*
1	90	5	0.2	4	0.01	50
2			0.4			25
3		10	0.2	6	0.00001	50
4			0.4			25
5	75	5	0.2	4	0.01	50
6			0.4			25
7		10	0.2	6	0.00001	50
8			0.4			25
9	40	5	0.2	4	0.01	50
10			0.4			25
11		10	0.2	6	0.00001	50
12			0.4			25
13	20	5	0.2	4	0.01	50
14			0.4			25
15		10	0.2	6	0.00001	50
16			0.4			25

Tabel 2: Hasil uji coba data IRIS

Skenario	*%Akurasi pembelajaran*	*%Akurasi pengujian*
IRIS_1	92.59	93.33
IRIS_2	92.74	94.67
IRIS_3	92.59	93.33
IRIS_4	91.85	97.58
IRIS_5	91.58	97.43
IRIS_6	92.10	97.43
IRIS_7	92.10	97.43
IRIS_8	92.10	97.43
IRIS_9	90.47	94.94
IRIS_10	90.47	94.71
IRIS_11	90.47	94.94
IRIS_12	90.47	94.40
IRIS_13	90.00	91.83
IRIS_14	89.33	91.16
IRIS_15	90.00	91.83
IRIS_16	88.89	94.83

Tabel 3: Hasil uji coba data LENSA

Skenario	*%Akurasi pembelajaran*	*%Akurasi pengujian*
LENSA_1	94.16	100.00
LENSA_2	95.83	100.00
LENSA_3	95.83	100.00
LENSA_4	95.83	100.00
LENSA_5	90.00	100.00
LENSA_6	91.11	100.00
LENSA_7	91.11	100.00
LENSA_8	90.00	100.00
LENSA_9	100.00	79.99
LENSA_10	100.00	85.41
LENSA_11	100.00	85.41
LENSA_12	100.00	87.22
LENSA_13	100.00	90.45
LENSA_14	100.00	89.47
LENSA_15	100.00	90.53
LENSA_16	100.00	91.58

Tabel 4: Hasil uji coba data WINE

Skenario	*%Akurasi pembelajaran*	*%Akurasi pengujian*
WINE_1	94.34	100.00
WINE_3	95.85	100.00
WINE_5	94.52	98.60
WINE_7	94.07	99.07
WINE_9	96.81	97.80
WINE_11	96.06	97.20
WINE_13	92.22	95.21
WINE_15	94.99	95.21

Tabel 5: Hasil uji coba data GLASS

Skenario	*%Akurasi pembelajaran*	*%Akurasi pengujian*
GLASS_1	59.63	75.00
GLASS_3	60.94	70.45
GLASS_5	62.57	66.55
GLASS_7	63.89	62.26
GLASS_9	69.46	57.74
GLASS_11	75.58	57.22
GLASS_13	71.81	58.15
GLASS_15	76.74	57.50

Hasil uji coba untuk data IRIS, LENSA, WINE dan GLASS diperlihatkan pada Tabel 2, Tabel 3, Tabel 4 dan Tabel 5 dengan skenario uji coba dari Tabel 1. Sebagai catatan hasil uji coba IRIS_1 pada Tabel 2 berarti pengujian dilakukan dengan variasi nilai – nilai parameter seperti yang ditunjukkan pada skenario 1 di Tabel 1. Analisa hasil uji coba menunjukkan bahwa pengurangan jumlah data pembelajaran tidak terlalu mempengaruhi tingkat akurasi pembelajaran dengan prosentase kebenaran sekitar 85% dan meningkat 5% saat dilakukan pengurangan data. Sedangkan tingkat akurasi pengujian mengalami penurunan ±10% dari 92.1% apabila jumlah data pembelajaran dikurangi. Pengamatan juga dilakukan pada jumlah iterasi yang dibutuhkan bagi algoritma genetik untuk mendapatkan nilai optimal dari *varians* ρ dan parameter regularisasi λ. Hasil pengamatan menunjukkan bahwa setidaknya iterasi sejumlah ±50 dibutuhkan untuk *varians* ρ mencapai nilai optimal dan sejumlah ±25 untuk parameter regularisasi λ. Hasil pengamatan

tersebut dilakukan pada pengujian data IRIS dengan % data pembelajaran = 90. Oleh karena itu variasi jumlah iterasi dipilih untuk 25 dan 50 kali seperti yang ditunjukkan pada skenario uji coba di Tabel 1. Dikarenakan kecendurangan nilai akurasi untuk % *crossover* = 0.2 terlihat meningkat sehingga pengujian data WINE dan GLASS hanya dilakukan pada nilai tersebut. Pengamatan selanjutnya dilakukan pada hasil percobaan dengan nilai %*crossover* = 0.2 dan jumlah data pembelajaran yang semakin berkurang. Hasil pengamatan menunjukkan bahwa adanya variasi nilai populasi, *crosspoint*, dan *threshold* tetap dapat menghasilkan nilai *varians* ρ dan parameter regularisasi λ sedemikian hingga yang membuat akurasi

klasifikasi tidak berubah seperti IRIS_1 – IRIS_3, IRIS_5 – IRIS_7, IRIS_9 – IRIS_11, IRIS_13 – IRIS_15, WINE_9 – WINE_11, WINE_13 – WINE_15. Pengamatan terakhir dilakukan berdasarkan hasil skenario terakhir yaitu 1 – 3 – 5 – 7 – 9 – 11 – 13 – 15. Tingkat akurasi klasifikasi dari skenario 1 – 3 – 5 – 7 sedikit lebih baik, ±4%, daripada skenario lainnya. Kemudian pengujian dilakukan pada data BREAST berisi 699 data penyakit kanker payudara 9 fitur dengan kelas normal – tidak normal (kanker). Hasil %akurasi klasifikasi adalah 94.16, 95.83, 90.00, 91.11 untuk BREAST_1, BREAST_3, BREAST_5, BREAST_7. Berdasarkan uji coba klasifikasi data BREAST dapat dikatakan bahwa sebaiknya data pembelajaran tidak kurang dari 50% karena hasil BREAST_3 lebih baik dari BREAST_5. Jadi dengan data pembelajaran > 50% maka akurasi klasifikasi tidak menurun meskipun dilakukan variasi set parameter genetik (populasi, *crosspoint*, dan *threshold*) karena nilai *varians* ρ dan parameter regularisasi λ untuk model pembelajaran

pada persamaan (10) dan (11) cenderung menghasilkan tingkat akurasi sama.

Secara umum uji coba klasifikasi dengan nilai bobot dalam arsitektur RBFNN untuk data IRIS, data LENSA, dan data WINE rata-rata mempunyai akurasi 88% sampai 100%, baik pada pembelajaran maupun pengujian. Sedangkan untuk data GLASS mempunyai rata-rata akurasi diatas 57% sampai 80%. Hasil akurasi tersebut didapatkan dengan pembelajaran nilai optimal untuk varians $\rho > 0.5$ dan parameter regularisasi $\lambda < 0.1$. Parameter regularisasi yang dihasilkan untuk pembelajaran mempunyai kecenderungan nilai dibawah 0.1 sedemikian hingga pembelajaran dengan arsitektur RBFNN untuk kesemua data tersebut dapat menggunakan pendekatan OLS yang tidak teregularisasi. Pengujian algoritma pembelajaran bertingkat pada makalah ini belum dapat menunjukkan penggunaan ROLS sebagai alternatif lebih baik daripada OLS. Hal ini mungkin disebabkan karena skala data uji coba terlalu kecil dengan jumlah data < 250 dan dataset tersebut tidak banyak memiliki data bersifat *noise*.

5. PENUTUP

Metode pembelajaran dengan cara kombinasi untuk pengklasifikasian dengan arsitektur jaringan saraf fungsi radial basis (*Radial Basis Function Neural Network*, RBFNN) telah dijelaskan dalam makalah ini. Metode kombinasi tersebut adalah *Regularized Orthogonal Least Squares* (ROLS) yang menjamin adanya generalisasi pada pembelajaran dikarenakan kemungkinan adanya data *noise* terlalu tinggi. Untuk menemukan nilai parameter regularisasi yang optimal maka pembelajaran dikombinasikan dengan Algoritma Genetik. Akan tetapi pengujian algoritma pembelajaran bertingkat pada makalah ini belum dapat menunjukkan penggunaan ROLS sebagai alternatif lebih baik. Hal tersebut mungkin disebabkan karena skala data uji coba terlalu kecil sehingga dataset tidak banyak memiliki data bersifat *noise*.

DAFTAR PUSTAKA

[1] M. Orr, "Introduction to Radial Basis Function Networks", *unpublished*, 1996.
url: www.anc.ed.ac.uk/~mjo/papers/intro.ps

[2] S. Chen, X. Hong, C. Harris, and L. Hanzo, "Fully Complex-valued Radial Basis Function Networks: Orthogonal Least Squares Regression and Classification", in *Neurocomputing*, vol. 71, no. 16 – 18, pp. 3421–3433, 2008.

[3] S. Chen, Y. Wu, and B. Luk, "Combined Genetic Algorithm Optimization and Regularized Orthogonal Least Squares Learning for Radial Basis Function Networks", in *IEEE Trans. on Neural Networks*, vol. 10, no. 5, 1239–1243, 1999.

[4] S. Chen, X. Hong, B. Luk, and C. Harris, "Orthogonal Least Squares Regression: A Unified Approach for Data Modelling", in *Neurocomputing*, vol. 72, no. 10 – 12, pp. 2670–2681, 2009.

[5] D. Whitley, "A Genetic Algorithm Tutorial", in *Statistics and Computing*, vol. 4, no. 2, 65–85, 1994.

[6] S. Bay, D. Kibler, M. Pazzani, and P. Smyth, "The UCI KDD Archive of Large Data Sets for Data Mining Research and Experimentation", in *ACM SIGKDD Explorations*, vol. 2, no. 2, 81–85, 2000.