
Perbandingan Model Estimasi *Artificial Neural Network* Optimasi *Genetic Algorithm* dan Regresi Linier Berganda

Jimmy Saputra Sebayang¹, Budi Yuniarto²

^{1,2}Jurusan Komputasi Statistik, Sekolah Tinggi Ilmu Statistik (STIS)

Email : ¹ jimmys@bps.go.id, ² byuniarto@stis.ac.id

DOI: 10.14710/medstat.10.1.13-23

Article Info:

Received: 12 Agustus 2016

Accepted: 14 Juni 2017

Available Online: 14 Agustus 2017

Keywords:

Neural Network, Genetic Algorithm, Ordinary Least Square

Abstract: Multiple Linear Regression is a statistical approach most commonly used in performing predictive data modeling. One of the methods that can be used in estimating the parameters of the model on Multiple Linear Regression is Ordinary Least Square. It has classical assumptions requirements and often the assumptions are not satisfied. Another method that can be used as an alternative data modeling is Artificial Neural Network. It is a free-distribution estimator because there's no assumptions that have to be satisfied. However, modeling data using ANN has some problems such as selection of network topology, learning parameters and weight initialization. Genetic Algorithm method can be used to solve those problems. A set of simulation data was generated to test the reliability of ANN-GA model compared to Multiple Linear Regression model. Model comparison experiments indicate that ANN-GA model are better than Multiple Linear Regression model for estimating simulation data both on the data training and data testing.

1. PENDAHULUAN

Dalam melakukan pemodelan data prediktif, teknik Regresi Linier Berganda merupakan pendekatan statistik yang paling umum dan teknik yang paling populer^[1]. Untuk mengestimasi parameter model pada teknik Regresi Linier Berganda tersebut, metode *Ordinary Least Square* (OLS) adalah salah satu metode yang dapat digunakan dengan kelebihanannya sebagai penduga yang tidak bias dan memiliki varians yang minimum diantara semua penduga linier yang tidak bias (Kutner, *et al.*, 2005).

Namun penggunaan teknik Regresi Linier Berganda dengan metode penduga *Ordinary Least Square* (OLS) memiliki persyaratan asumsi-asumsi yang mendasarinya dan sering ditemukan bahwa data yang akan dimodelkan tidak memenuhi asumsi-asumsi klasik yang disyaratkan tersebut. Jika data yang tidak memenuhi asumsi-asumsi klasik tersebut tetap dimodelkan dengan menggunakan pendekatan tersebut, maka akan terjadi pelanggaran asumsi yang akan mengakibatkan hasil yang diperoleh menjadi jauh dari harapan, tidak efisien, tidak konsisten atau bahkan menjadi bias.

Salah satu metode alternatif yang dapat digunakan untuk menangani permasalahan tersebut adalah dengan menggunakan metode *Artificial Neural Network*. Metode *Artificial Neural Network* merupakan metode *soft computing* yang *powerful* dalam melakukan pemodelan data dengan cara prediktif. Metode tersebut merupakan model estimasi bebas karena tidak bergantung pada bentuk asumsi yang mendasari data. Sehingga, metode ini dapat digunakan sebagai alternatif pemodelan jika asumsi-asumsi pada metode klasik tidak terpenuhi. Namun penggunaan metode *Artificial Neural Network* membutuhkan pemilihan topologi jaringan serta paramater pembelajaran yang akan digunakan dalam proses pembelajaran. Pemilihan topologi dan paramater tersebut merupakan salah satu faktor kesuksesan pembelajaran ANN. Penentuan nilai parameter dan topologi jaringan yang buruk akan mempengaruhi kinerja ANN serta model yang dihasilkan.

Diluar penentuan nilai paramater dan topologi jaringan, permasalahan lain yang muncul ketika melakukan pemodelan dengan menggunakan metode *Artificial Neural Network* adalah penentuan inisiasi bobot awal jaringan yang akan digunakan dalam pembelajaran (*training*). Leondes (1998) menuliskan bahwa nilai inisiasi bobot dapat secara dramatis berefek terhadap perilaku pembelajaran. Jika nilai inisiasi bobot buruk, akan membutuhkan waktu yang lama untuk memperoleh konvergensi yang memenuhi syarat. Bahkan kemungkinan terburuknya, jaringan mungkin akan terjebak pada sebuah minimum lokal yang buruk.

Untuk menyelesaikan permasalahan tersebut, metode *Genetic Algorithm* dapat digunakan dengan kemampuannya dalam menyelesaikan masalah untuk pencarian solusi yang optimal. Gen dan Cheng (1997) menuliskan bahwa Algoritma ini memungkinkan digunakan pada optimasi masalah dengan ruang pencarian yang sangat luas dan kompleks serta memungkinkan untuk mendapatkan nilai optimum global (yang terhindar dari daerah optimum lokal).

Berdasarkan uraian permasalahan diatas, penelitian ini bertujuan untuk membandingkan model *Artificial Neural Network* optimasi *Genetic Algorithm* dan Regresi Linier Berganda yang difokuskan pada analisis tingkat akurasi estimasi yang dihasilkan dari kedua model tersebut. Sekumpulan data simulasi dibangkitkan dengan mengikuti distribusi normal yang digunakan sebagai studi kasus untuk menguji kehandalan kedua model tersebut.

Pada penelitian ini, pendekatan topologi *Artificial Neural Network* yang digunakan adalah *multilayer feed-forward* dengan algoritma pembelajaran *backpropagation*. *Genetic Algorithm* digunakan untuk pemilihan parameter pembelajaran (momentum dan *learning rate*) serta penginisiasian bobot awal jaringan yang optimum. Ukuran yang digunakan untuk menilai seberapa baik model yang dihasilkan menggunakan besaran nilai *r-square* dan RMSE (*Root Mean Square Error*).

2. TINJAUAN PUSTAKA

2.1 *Backpropagation-Artificial Neural Network*

Metode *Artificial Neural Network* yang digunakan pada penelitian ini menggunakan algoritma pembelajaran *backpropagation* yang diperluas dengan penambahan konstanta momentum α pada parameter pembelajarannya dan fungsi aktivasi sigmoid (*fermi function*) pada semua neuron. Penambahan momentum dilakukan atas pendapat Kriesel (2007) yang menyatakan bahwa parameter momentum berguna untuk memperoleh nilai turunan gradien (*gradien descent*) yang konvergen. Sementara penggunaan fungsi aktivasi sigmoid (*fermi function*) pada neuron didasarkan atas pendapat Rojas (1996) dan Kriesel (2007) yang menyebutkan bahwa fungsi aktivasi tersebut merupakan fungsi aktivasi yang paling

populer untuk pembelajaran *backpropagation*. Fungsi aktivasi sigmoid ditunjukkan pada Persamaan (1).

$$S(x) = \frac{1}{1+e^{-x}} \quad , \quad S : R \rightarrow (0,1) \quad (1)$$

Adapun pembaharuan bobot pada jaringan dengan penambahan momentum pada pembelajaran *backpropagation* didefinisikan pada Persamaan (2).

$$\Delta w_{ij}(t) = -\gamma \frac{\partial E}{\partial w_{ij}} + \alpha \Delta w_{ij}(t-1) \quad (2)$$

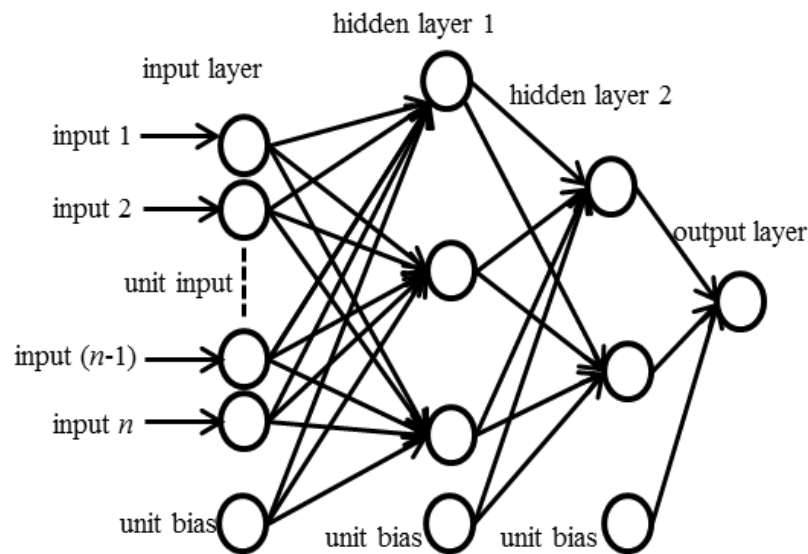
Nilai $\Delta w_{ij}(t)$ merupakan besaran perubahan bobot pada neuron ke- i , sinapsis ke- j pada iterasi ke- t dan γ adalah nilai parameter *learning rate*. Sementara $\frac{\partial E}{\partial w_{ij}}$ merupakan besaran turunan parsial dari galat yang didefinisikan pada Persamaan (3) untuk *output layer* dan Persamaan (4) untuk *hidden layer* (Rojas, 1996).

$$\frac{\partial E}{\partial w_{ij}} = [o_j (1 - o_j) (o_j - t_j)] o_i \quad (3)$$

$$\frac{\partial E}{\partial w_{ij}} = \delta_j o_i = [o_j (1 - o_j) (\sum_{q=1}^{m+1} w_{jq} \cdot \delta_m)] o_i \quad (4)$$

o_i merupakan besaran *output* yang dihasilkan neuron ke- i , t adalah besaran nilai *target* dan δ merupakan besaran galat *backpropagation* (delta).

Alur proses pelatihan jaringan pada metode *Artificial Neural Network* dalam penelitian ini ditunjukkan pada Gambar 2. Contoh sederhana arsitektur *Artificial Neural Network* yang digunakan pada penelitian ini ditunjukkan pada Gambar 1.

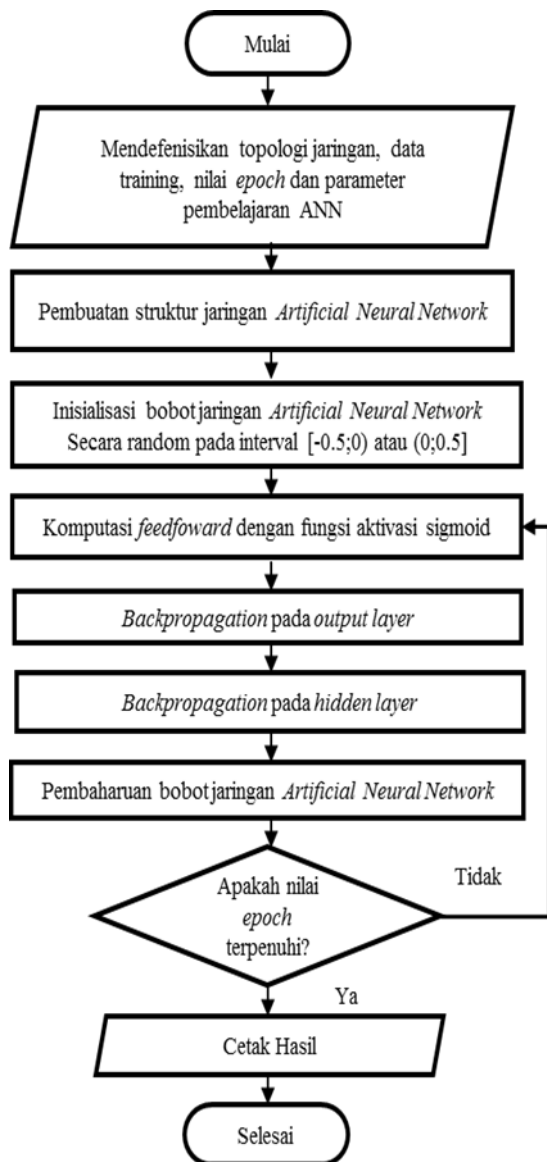


Gambar 1. Arsitektur ANN

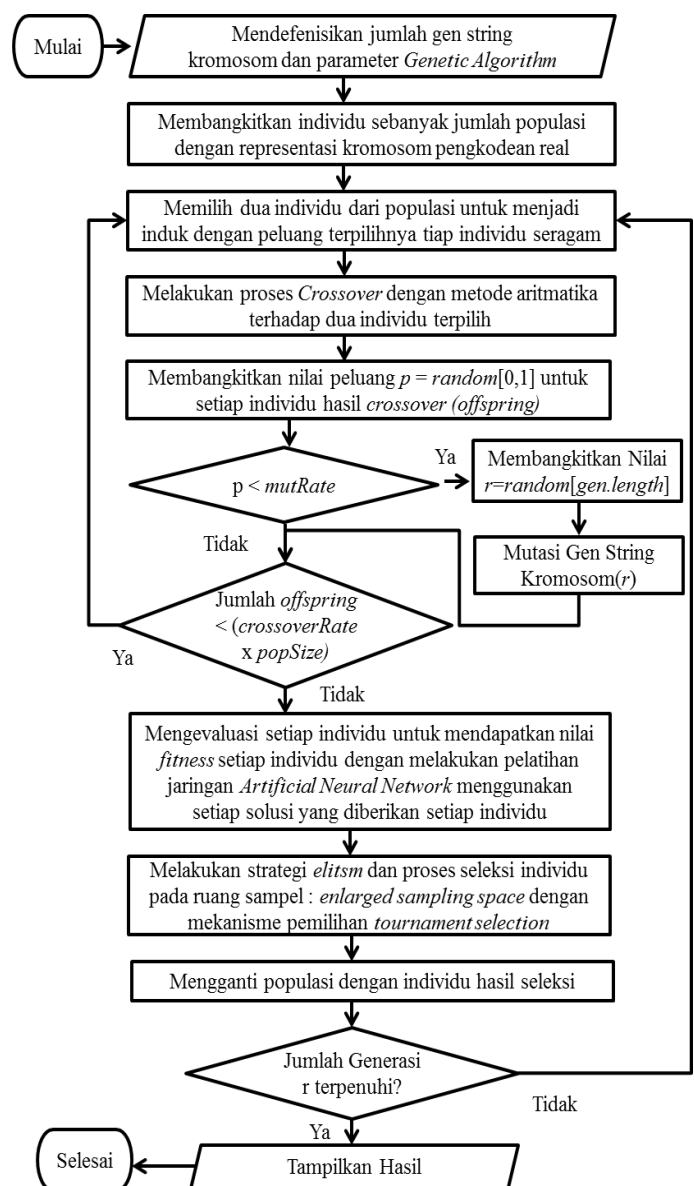
2.2 Genetic Algorithm

Siklus *Genetic Algorithm* dimulai dengan membuat populasi awal yang terdiri dari individu-individu yang dibangkitkan secara acak (Gen & Cheng (1997), Herera, *et al.* (1998), dan Michalewicz (1996)). Individu-individu tersebut didefinisikan sebagai salah satu solusi yang mungkin dari kombinasi nilai *learning rate*, momentum dan inisialisasi

bobot. Solusi atas permasalahan tersebut langsung direpresentasikan menjadi gen string kromosom dengan menggunakan teknik pengkodean real (*Real-Coded GA/RCGA*).



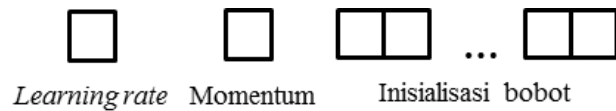
Gambar 2. Alur proses pelatihan ANN



Gambar 3. Alur Siklus Evolusi GA

Penggunaan metode *Genetic Algorithm* pengkodean real dipilih karena teknik pengkodean biner sulit dilakukan ketika daerah pencarian berada pada daerah kontinu dengan dimensi yang besar serta dibutuhkan angka presisi yang baik. Penggunaan GA dengan pengkodean biner juga memerlukan banyak waktu komputasi pada optimasi fungsi yang kompleks karena transformasi dari bilangan desimal (*real*) ke bilangan biner dan sebaliknya. Penggunaan GA dengan pengkodean real dapat mengatasi masalah ini karena memungkinkan untuk bekerja pada daerah/domain yang luas pada variabel kontinu. Keuntungan lain ketika menggunakan pengkodean *real* adalah kemampuannya untuk mengeksplorasi secara berangsur-angsur pada fungsi dengan variabel kontinu (Herera, *et al.*, 1998). Titik permasalahan nilai *learning rate*, momentum dan inisialisasi bobot ANN yang akan diselesaikan berada dalam daerah kontinu serta jumlah solusi atas permasalahan

inisialisasi bobot ANN yang akan dicari tidaklah sedikit. Oleh karena itu, teknik pengkodean real dipilih untuk merepresentasikan kromosom. Representasi gen string kromosom individu GA tersebut ditunjukkan pada Gambar 4.



Gambar 4. Representasi kromosom

Pada proses pencarian parameter pembelajaran *backpropagation-Artificial Neural Network* pada gen string kromosom pertama dan kedua, ruang pencarian akan dibatasi berdasarkan pendapat Kriesel (2007) yang menuliskan bahwa pengalaman sebelumnya menunjukkan bahwa nilai *learning rate* yang baik berada pada rentang 0,01 sampai 0,9. Sementara untuk nilai momentum berada pada rentang 0,6 sampai 0,9. Sementara untuk gen string kromosom permasalahan inisialisasi bobot pada *Artificial Neural Network*, ruang pencarian akan mengikuti solusi sederhana yang disarankan Kriesel (2007) yaitu menggunakan *symmetry breaking* dengan melakukan inisialisasi bobot pada jaringan dengan nilai-nilai yang kecil dan dipilih secara acak. Rentang nilai acak berada pada interval [-0.5;0] atau (0;0.5].

Panjang gen string kromosom pada individu yang ditunjukkan pada Gambar 3 tersebut akan bersifat dinamis. Besarnya jumlah kromosom tersebut tergantung pada jumlah lapisan tersembunyi dan jumlah neuron masing-masing lapisan tersembunyi yang telah diperoleh dari hasil pencarian. Jumlah-jumlah tersebut akan mendefinisikan seberapa banyak sinapsis (interkoneksi antar neuron) yang terbentuk pada topologi. Berdasarkan percobaan yang telah dilakukan, Persamaan (5) dapat digunakan untuk menghitung jumlah sinapsis tersebut yang juga mendefinisikan besaran gen string kromosom inisialisasi bobot.

$$s = \sum_{i=1}^{l-1} (n_i + b_i) (n_{i+1}) \quad (5)$$

s adalah jumlah sinapsis (interkoneksi antar neuron) yang terbentuk pada jaringan yang juga merupakan jumlah gen string kromosom untuk inisialisasi bobot, n_i adalah jumlah neuron pada lapisan ke- i (dimulai dari lapisan masukan sampai pada lapisan keluaran tidak termasuk neuron bias didalamnya) dan b merupakan jumlah neuron bias pada lapisan ke- i (yang biasanya berjumlah 1 pada setiap *hidden layer* dan *input layer*).

Setelah membuat populasi awal (inisialisasi), tahap selanjutnya adalah melakukan proses reproduksi yang terdiri dari operator *crossover* dan operator mutasi (Gen & Cheng, 1997). Pada proses *crossover*, individu-individu yang dipilih menjadi induk (*parent*) akan dilakukan secara acak (seragam) yang artinya peluang setiap individu dalam populasi untuk terpilih adalah sama. Metode tukar silang (*crossover*) yang digunakan adalah metode *crossover* aritmatika yang dituliskan pada Persamaan (6a) dan (6b).

$$z_1 = \alpha x_1 + (1 - \alpha)x_2 \quad (6a)$$

$$z_2 = \alpha x_2 + (1 - \alpha)x_1 \quad (6b)$$

x merupakan gen string kromosom orang tua terpilih, z adalah gen string kromosom individu *offspring* yang dihasilkan dan α merupakan bilangan random yang dipilih secara acak pada interval [0,1] (Michalewicz, 1994).

Adapun nilai tingkat *crossover* didefinisikan mengikuti pendapat Gen & Cheng (1997) yang menyatakan bahwa nilai tersebut yang dilambangkan dengan p_c dituliskan sebagai rasio keturunan yang dihasilkan proses *crossover* pada setiap generasi terhadap

ukuran populasi sehingga akan dihasilkan keturunan sebanyak nilai tingkat *crossover* dikalikan dengan ukuran populasi.

Setelah proses *crossover* dilakukan, proses mutasi akan dilakukan terhadap kedua anak hasil tukar silang (*crossover*) dengan peluang dilakukannya mutasi sebesar tingkat mutasi (*mutation rate*). Mutasi dilakukan dengan menetapkan sebuah bilangan random yang dibangkitkan dari distribusi seragam pada domain pencarian yang dituliskan pada Persamaan (7) (Simon, 2013).

$$\begin{aligned} r &\rightarrow U[0,1] \\ x_i(k) &\rightarrow U[x_{min}(k), x_{max}(k)] \text{ jika } r < \rho \end{aligned} \quad (7)$$

$i \in [1, N]$ dan $k \in [1, n]$, ρ adalah nilai tingkat mutasi, N adalah jumlah individu dalam populasi dan n adalah jumlah string gen kromosom dalam individu.

Pada tahap selanjutnya dalam siklus *Genetic Algorithm* adalah proses evaluasi. Individu-Individu yang ada dievaluasi menggunakan nilai *fitness* (Gen & Cheng, 1997). Fungsi *fitness* tersebut menggambarkan tingkat optimalitas dari sebuah individu (Gorunescu, 2011). Dalam penelitian ini, fungsi *fitness* tersebut akan diukur dengan melihat seberapa besar tingkat akurasi model jaringan ANN yang diberikan oleh masing-masing individu menggunakan ukuran NRMSE (*Normalized Root Mean Square Error*) yang dituliskan pada Persamaan (8).

$$f(x) = 1 - x, \quad 0 \leq x \leq 1 \quad (8)$$

$f(x)$ merupakan nilai *fitness* yang dihasilkan dari sebuah individu GA dan x adalah nilai RMSE jaringan ANN (NRMSE) yang dihasilkan dari sebuah individu GA.

Setelah semua individu dievaluasi, tahapan selanjutnya adalah proses seleksi dan strategi *elitist* (*elitism*). Proses seleksi pada penelitian ini mengacu pada pemilihan individu untuk membentuk populasi baru pada generasi berikutnya. Sementara strategi *elitist*, Gen & Cheng (1997); Gorunescu (2011) menuliskan digunakan untuk menghindari hilangnya individu terbaik akibat tidak terpilihnya dalam proses seleksi. Adapun ruang sampel yang digunakan dalam pemilihan tersebut adalah *enlarged sampling space*, baik orang tua dan anak yang dihasilkan (*offspring*) memiliki kesempatan yang sama untuk bertahan ke generasi berikutnya. Mekanisme pemilihan tersebut dilakukan dengan menggunakan metode seleksi *tournament*.

Alur proses metode *Genetic Algorithm* dalam penelitian ini ditunjukkan pada Gambar 3. Sementara untuk nilai standar parameter GA, pada penelitian ini diberikan jumlah populasi = 60 ;crossover rate = 0.25; mutation rate = 0.01; dan jumlah generasi = 1000. Penentuan nilai standar tersebut diadaptasi dari penelitian Michalewicz (1996). Pada penelitian tersebut, jumlah generasi (iterasi) ditentukan sebanyak 20000. Namun untuk menghemat waktu komputasi, jumlah generasi ditentukan sebanyak 1000.

2.3 Pencarian Topologi Jaringan ANN

Dalam penelitian ini, pencarian jumlah *hidden layer* dan jumlah neuron pada masing-masing *hidden layer* tidak tercakup pada penggunaan metode *Genetic Algorithm*. Pencarian nilai-nilai tersebut akan dilakukan melalui algoritma pencarian topologi jaringan yang diajukan dalam penelitian ini dengan mengadaptasi batasan pencarian yang disarankan oleh Heaton (2008). Hasil pencarian pada algoritma ini akan menjadi dasar pembentukan gen string kromosom GA.

Dari berbagai aturan metode praktis yang dapat digunakan sebagai acuan dalam pemilihan jumlah neuron pada *hidden layer*, Heaton (2008) meringkas aturan tersebut sebagai berikut :

- Jumlah neuron pada *hidden layer* sebaiknya berada diantara ukuran *input layer* dan *output layer*.
- Jumlah neuron pada *hidden layer* sebaiknya berjumlah $2/3$ dari ukuran *input layer* ditambah dengan ukuran *output layer*.
- Jumlah neuron pada *hidden layer* sebaiknya kurang dari dua kali ukuran *input layer*.

Selanjutnya dari ketiga aturan praktis tersebut, dalam penelitian ini diringkas menjadi suatu himpunan daerah pencarian jumlah neuron yang dituliskan pada Persamaan 9.

$$H = \{ x \mid 1 < x < 2n, x, n \in N \} \quad (9)$$

H merupakan himpunan daerah pencarian jumlah neuron, x adalah jumlah neuron hasil pencarian dan n adalah jumlah neuron pada *input layer*.

Untuk penentuan jumlah *hidden layer*, *Artificial Neural Network* dengan dua *hidden layer* dapat merepresentasikan fungsi dengan berbagai macam bentuk dan belum terdapat alasan teoritis untuk menggunakan model *Artificial Neural Network* dengan lebih dari dua *hidden layer*. Pada beberapa masalah praktis, tidak ada alasan menggunakan model *Artificial Neural Network* dengan jumlah *hidden layer* lebih dari satu (Heaton, 2008). Dengan demikian, hasil pencarian jumlah *hidden layer* pada penelitian ini akan berjumlah satu atau dua *layer*.

Setelah menentukan batasan pencarian, selanjutnya sebanyak k individu GA yang direpresentasikan pada Gambar 4 untuk setiap kemungkinan solusi topologi jaringan (kombinasi jumlah *hidden layer* dan neuron per *hidden layer*) dibangkitkan secara random. Setiap individu tersebut kemudian dievaluasi dengan menghitung nilai *fitness*-nya. Nilai rata-rata *fitness* individu-individu pada setiap kemungkinan solusi topologi jaringan dihitung. Nilai rata-rata *fitness* tersebut merupakan ukuran seberapa baik solusi topologi jaringan yang dihasilkan dan seberapa baik individu-individu GA dalam populasi awal yang akan dibangkitkan dengan menggunakan topologi jaringan tersebut. Semakin besar nilai rata-rata *fitness* yang didapatkan, semakin baik individu-individu solusi GA yang akan diperoleh.

2.4 Regresi Linier Berganda

Regresi Linier Berganda adalah salah satu teknik analisis data dalam metode statistik yang sering digunakan untuk mengkaji hubungan/pengaruh antara beberapa variabel respon terhadap suatu variabel prediktor dan juga digunakan untuk keperluan memprediksi suatu variabel. Bentuk umum model Regresi Linier Berganda dengan variabel prediktor sebanyak $(p - 1)$ dituliskan pada Persamaan 9. Salah satu metode yang dapat digunakan untuk mengestimasi parameter yang terdapat pada persamaan Regresi Linier Berganda tersebut adalah metode *Ordinary Least Square*. Metode tersebut meminimumkan jumlah kuadrat error yang dihasilkan oleh model (Kutner, *et al.*, 2005).

$$Y_i = \beta_0 + \sum_{k=1}^{p-1} \beta_k X_{ik} + \varepsilon_i \quad (10)$$

2.5 Data Simulasi

Pada penelitian ini, data yang digunakan merupakan data simulasi yang dibangkitkan mengikuti distribusi normal yang terdiri dari 3 komponen variabel input dan 1 komponen variabel error dengan jumlah observasi bangkitan sebanyak 150.

2.6 Metode Analisis

Metode analisis yang digunakan adalah Uji Tanda (*Sign Test*) dengan melakukan dua pengujian statistik yakni pengujian perbandingan antara metode ANN-GA dengan metode Regresi Linier Berganda untuk data training dan untuk data testing. Pengujian tersebut akan dilakukan sebanyak 10 kali percobaan dengan membagi data simulasi menjadi dua bagian yaitu bagian data training dan bagian data testing dengan komposisi 80% dan 20%. Pemilihan observasi yang akan menjadi data training atau data testing tersebut akan dilakukan secara acak (seragam). Cara pemilihan tersebut dilakukan berdasarkan penelitian Yeh (1998) yang menemukan bahwa pemilihan dengan menggunakan *random sampling* memberikan hasil yang lebih baik. Besaran α yang digunakan adalah 0.05 dengan hipotesis yang digunakan sebagai berikut :

H_0 : Hasil estimasi model ANN-GA memberikan hasil yang sama jika dibandingkan dengan model Regresi Linier Berganda.

H_1 : Hasil estimasi model ANN-GA memberikan hasil yang lebih baik jika dibandingkan dengan model Regresi Linier Berganda.

Tabel 1. Nilai *r-square* Hasil Percobaan Pemodelan Data Simulasi dengan Metode *Artificial Neural Network-Genetic Algorithm* dan Regresi Linier Berganda

Percobaan	Data Training			Data Testing		
	ANN-GA	RLB	sign	ANN-GA	RLB	sign
1	0.93037	0.90818	+	0.92806	0.92272	+
2	0.93189	0.91126	+	0.92847	0.91833	+
3	0.93809	0.9206	+	0.92033	0.88957	+
4	0.91712	0.90758	+	0.93213	0.94113	-
5	0.93371	0.91507	+	0.91155	0.92287	-
6	0.92162	0.91415	+	0.92054	0.90231	+
7	0.93894	0.92609	+	0.85579	0.84365	+
8	0.92287	0.91722	+	0.91458	0.90987	+
9	0.90245	0.89628	+	0.94851	0.95984	-
10	0.93496	0.92351	+	0.87755	0.88607	-

Pada penelitian ini, pengambilan keputusan dilakukan dengan menggunakan pendekatan binomial karena jumlah sampel yang digunakan adalah sebanyak 10. Himpunan daerah penolakan dalam penelitian ini terdiri dari semua harga x (banyaknya tanda kurang) dibawah 5 ($x \leq 4$). Tanda kurang tersebut menyatakan bahwa hasil pemodelan metode ANN-GA memberikan nilai *R-square* yang tidak lebih baik dari pendekatan statistik teknik Regresi Linier Berganda. Ukuran *R-square* tersebut dipilih berdasarkan penelitian Yeh (1998) yang membandingkan performa dari *Artificial Neural Network* dengan teknik regresi dengan mengadopsi ukuran tersebut. Selain *R-square*, ukuran lain yang digunakan untuk mendukung hasil dalam penelitian ini adalah RMSE (*Root Mean Square Error*). Comrie (2012) menggunakan ukuran tersebut sebagai salah

satu pembandingan model dalam penelitiannya. H_0 akan ditolak jika nilai x (banyaknya tanda kurang) yang diperoleh berada pada daerah penolakan ($x \leq 4$).

3. HASIL DAN PEMBAHASAN

Hasil percobaan yang telah dilakukan baik pada data *training* maupun data *testing* masing-masing percobaan menggunakan metode ANN-GA dan teknik RLB, diperoleh nilai *r-square* yang ditunjukkan pada Tabel 1 dan nilai RMSE ditunjukkan pada Tabel 2. Model *backpropagation-Artificial Neural Network* yang digunakan dalam masing-masing percobaan tersebut ditunjukkan pada Tabel 3 yang merupakan hasil optimasi pencarian metode *Genetic Algorithm* dan hasil pencarian topologi jaringan. Sementara untuk teknik *Ordinary Least Square-Regresi Linier Berganda*, masing-masing model yang dihasilkan pada masing-masing percobaan ditunjukkan pada Tabel 4.

Tabel 2. Nilai RMSE (*Root Mean Square Error*) Hasil Percobaan Pemodelan Data Simulasi dengan Metode ANN-GA dan Regresi Linier Berganda

Percobaan	Data Training			Data Testing		
	ANN-GA	RLB	sign	ANN-GA	RLB	sign
1	1.83533	2.10731	+	2.24246	2.13079	-
2	1.86912	2.14704	+	1.89544	2.04417	+
3	1.76648	1.99973	+	2.42339	2.63018	+
4	2.12245	2.23563	+	1.63609	1.55287	-
5	1.8195	2.05947	+	2.33589	2.39096	+
6	2.10459	2.20007	+	1.6216	1.73999	+
7	1.79985	1.98116	+	2.44381	2.60981	+
8	2.00895	2.0794	+	2.12838	2.296	+
9	2.17864	2.2432	+	1.67072	1.4961	-
10	1.77542	1.91727	+	2.85998	2.82621	-

Tabel 3. Parameter Pembelajaran dan Topologi Jaringan *Artificial Neural Network* yang Digunakan dalam Percobaan Pemodelan Data

Percobaan	Learning Rate	Momentum	Hidden Layer	Unit Hidden	Epoch
1	0.18755	0.81111	2	5	4501
2	0.16122	0.63626	2	5	9930
3	0.25351	0.89513	1	5	7301
4	0.24525	0.89108	1	5	162
5	0.25892	0.84739	1	5	4235
6	0.27224	0.85776	2	5	180
7	0.25053	0.89595	1	5	8748
8	0.24339	0.88837	2	5	848
9	0.19641	0.89628	2	5	200
10	0.26306	0.89341	2	5	2863

Berdasarkan hasil percobaan tersebut, dari sepuluh percobaan yang dilakukan pada data *training*, metode ANN-GA memberikan nilai *r-square* (Tabel 1) yang lebih baik pada semua percobaan jika dibandingkan dengan metode RLB. Begitu juga dengan menggunakan ukuran RMSE pada Tabel 2, dari sepuluh percobaan yang telah dilakukan,

metode ANN-GA memberikan tingkat akurasi yang lebih baik (nilai RMSE yang diperoleh lebih kecil) pada semua percobaan jika dibandingkan dengan pendekatan statistik teknik Regresi Linier Berganda. Nilai x (jumlah tanda negatif) yang diperoleh pada percobaan tersebut baik menggunakan ukuran *r-square* maupun RMSE adalah 0 yang berarti nilai x yang diperoleh berada pada daerah penolakan. Dengan demikian, keputusan yang dapat diambil berdasarkan percobaan tersebut adalah menolak H_0 untuk hipotesis pada penelitian ini.

Tabel 4. Model Regresi Linier Berganda yang Dihasilkan pada Masing-Masing Percobaan

Percobaan	b_0	b_1	b_2	b_3
1	10.097	0.45992	0.46231	0.59241
2	9.09902	0.47808	0.46367	0.60055
3	7.13726	0.50133	0.48035	0.59068
4	9.92221	0.44067	0.46165	0.60777
5	7.07431	0.49455	0.47193	0.61385
6	10.3889	0.45746	0.45209	0.60718
7	8.72	0.447	0.477	0.597
8	7.635	0.465	0.476	0.613
9	10.3167	0.45311	0.45404	0.60768
10	9.194	0.48	0.451	0.62

Untuk hasil percobaan terhadap data *testing*, tidak semua percobaan memberikan hasil bahwa metode ANN-GA lebih baik dari metode RLB. Pada percobaan 4,5,9 dan 10, teknik RLB memberikan nilai *r-square* yang lebih baik untuk mengestimasi data di luar pembentukan model (*testing*). Sementara dengan menggunakan ukuran RMSE, percobaan 1,4,9,10 memberikan tingkat akurasi yang lebih baik pada teknik RLB. Namun nilai x (jumlah tanda negatif) tersebut yaitu sebanyak 4 masih berada pada daerah penolakan sehingga keputusan yang dapat diambil berdasarkan percobaan tersebut adalah menolak H_0 untuk hipotesis pada penelitian ini.

4. KESIMPULAN

Berdasarkan hasil percobaan baik pada data *training* maupun data *testing* dengan menggunakan data simulasi pada penelitian ini, memberikan keputusan untuk menolak H_0 penelitian ini. Oleh karena itu dapat ditarik sebuah kesimpulan dengan tingkat kepercayaan 95%, metode *Artificial Neural Network-Genetic Algorithm* memberikan tingkat akurasi yang lebih baik jika dibandingkan dengan metode Regresi Linier Berganda. Dengan demikian, metode *Artificial Neural Network-Genetic Algorithm* dapat digunakan sebagai alternatif pemodelan data jika berbagai asumsi klasik yang disyaratkan oleh teknik Regresi Linier Berganda tidak terpenuhi.

DAFTAR PUSTAKA

- Comrie, A.C., 2012. Comparing Neural Networks and Regression Models For Ozone Forecasting. *Journal of the Air and Waste Management Association*, Vol 47, No 6, 653-663. 8 Juli 2015.
<http://www.tandfonline.com/doi/abs/10.1080/10473289.1997.10463925>.
- Gen, M. & Cheng, R., 1997. *Genetic Algorithms and Engineering Design*. Canada : John Wiley and Sons, Inc.

- Gorunescu, F., 2011. Data Mining : Concepts, Models and Techniques. *Intelegent System Reference Library*, Vol 12. Jerman : Springer.
- Heaton, J., 2008. *Introduction Neural Networks with Java (2nd ed)*. USA : Heaton Research, Inc.
- Herrera, F., Lozano, M., & Verdegay, J. L., 1998. Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. *Artificial Intelegence Review*. Vol 12, No. 4, 265-319.
- Kriesel, D., 2007. *A Brief Introduction to Neural Networks*.
- Kutner, M.H., Nachtsheim, C.J., Neter, J., & Li, W., 2005. *Applied Linier Statistical Models (5th ed)*. New York : McGraw-Hill.
- Leondes, C.T., 1998. *Algorithms and Architectures*. USA :Academic Press.
- Michalewicz, Z., 1996. *Genetic Algorithms + Data Structures = Evolution Programs (3rd ed)*. New York : Springer-Verlag.
- Michalewicz, Z., 1994. Evolutionary Computation Techniques for Nonlinier Programming Problems. *International Transactions in Operational Research*. Vol 1, No. 2, 223-240.
- Rojas, R., 1996. *Neural Network : A System Introduction*. Jerman : Springer.
- Simon, D., 2013. *Evolutionary Optimization Algorithms*. Canada : John Wiley and Sons, Inc.
- Yeh, I.C., 1998. Modeling of Strength of High-Performance Concrete Using Artificial Neural Networks. *Cement and Concrete Research*, Vol 28, No.12, 1797-108.